CR-150586

Report No. 78-002
Contract No. NAS8-32047

T# 78-07290

N78-73149

(NASA-CR-150586) PAYLOAD SOFTWARE TECHNOLOGY. VOLUME 2: SOFTWARE TECHNOLOGY DEVELOPMENT PLAN Interim Final Report (M&S Computing, Inc., Huntsville, Ala.) 90 p

Unclas
00/16 07290

PAYLOAD SOFTWARE TECHNOLOGY

FINAL REPORT

VOLUME II

EXECUTIVE SUMMARY

AND

SOFTWARE TECHNOLOGY DEVELOPMENT PLAN

July 28, 1978

Prepared for:

George C. Marshall Space Flight Center
National Aeronautics and Space Administration
Marshall Space Flight Center, Alabama 35812

M&S COMPUTING, INC.

TABLE OF CONTENTS

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

TABLE OF CONTENTS
(Continued)

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The Software Technology Development Plan is the prime end-item of the Payload Software Technology Study (NASA Contract NAS8-32047). This plan was prepared by M&S Computing, Inc., for Mr. John Capps, COR, EF-15, Marshall Space Flight Center. The study encompassed four major software assessment tasks: (1) STS Sortie Payloads; (2) Spacelab Mission I; (3) supporting Ground-Based Software; and (4) Automated Payload Software. Each major task was composed of three study phases. Figure 1-1 describes the functional flow of the study. Figure 1-2 portrays the study schedule.

1.1 Purpose

The Software Technology Development Plan was formulated to define programmatic requirements for the advancement of Software Technology. The initial study was designed to identify drivers and suggest solutions to meet the requirements of space flight in the 1980-1990 time period, and to project potential technological needs from 1990 through the end of the century.

All identified software technology requirements were reviewed to establish the following: which requirements can be fulfilled within current state-of-the-art technology; which technology requirements can be developed with a high probability of success and a significant payoff in benefits; and which requirements pose risks (in cost and success) but are considered mandatory for other reasons. The resulting software technology development items are compiled into this development plan. The plan describes justification, cost and schedule, potential payoff benefits, measurable milestones, methods of approach, and methods for verification of results.

A large number of Software Technology Items were identified; however, it does not make sense for NASA to duplicate technology development which is being satisfactorily performed by industry or other government agencies. Therefore, this plan is dedicated to fresh ideas or known items which are not being worked on to the extent necessary. Goals have been established by the Outlook For Space Study (OFS) and the Office of Aeronautics and Space Technology (OAST) Workshops. This plan can only identify the next step toward these goals. Hopefully, it is a major step.

1.2 Scope

The primary emphasis in the Software Technology Development Plan is onboard processing in a real-time environment. Payload analysis was based on projected Space Transportation System (STS) payloads as described

PAYLOAD SOFTWARE TECHNOLOGY STUDY FUNCTIONAL FLOW

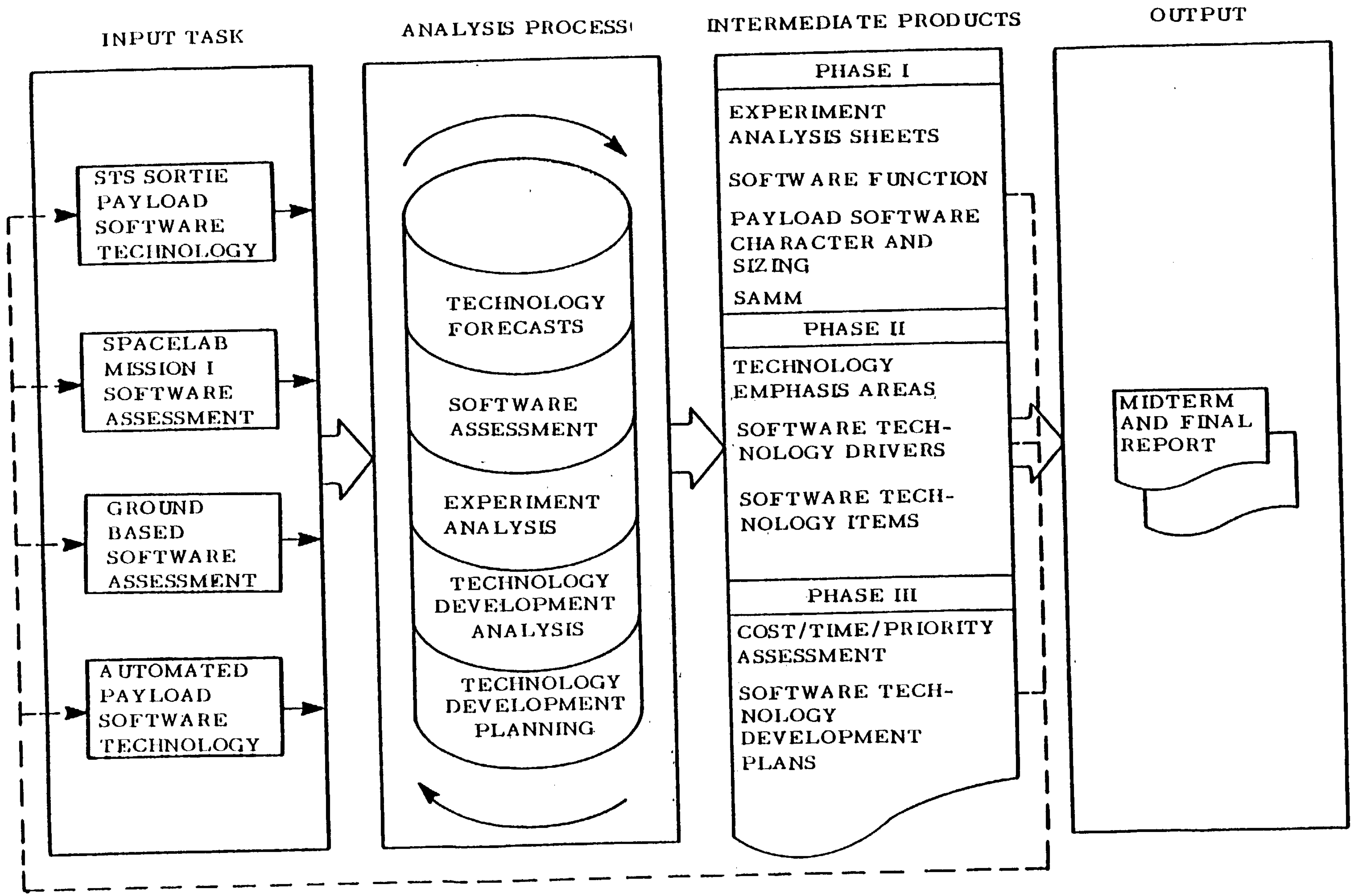


Figure 1-1

PAYLOAD SOFTWARE TECHNOLOGY
SCHEDULE OVERVIEW

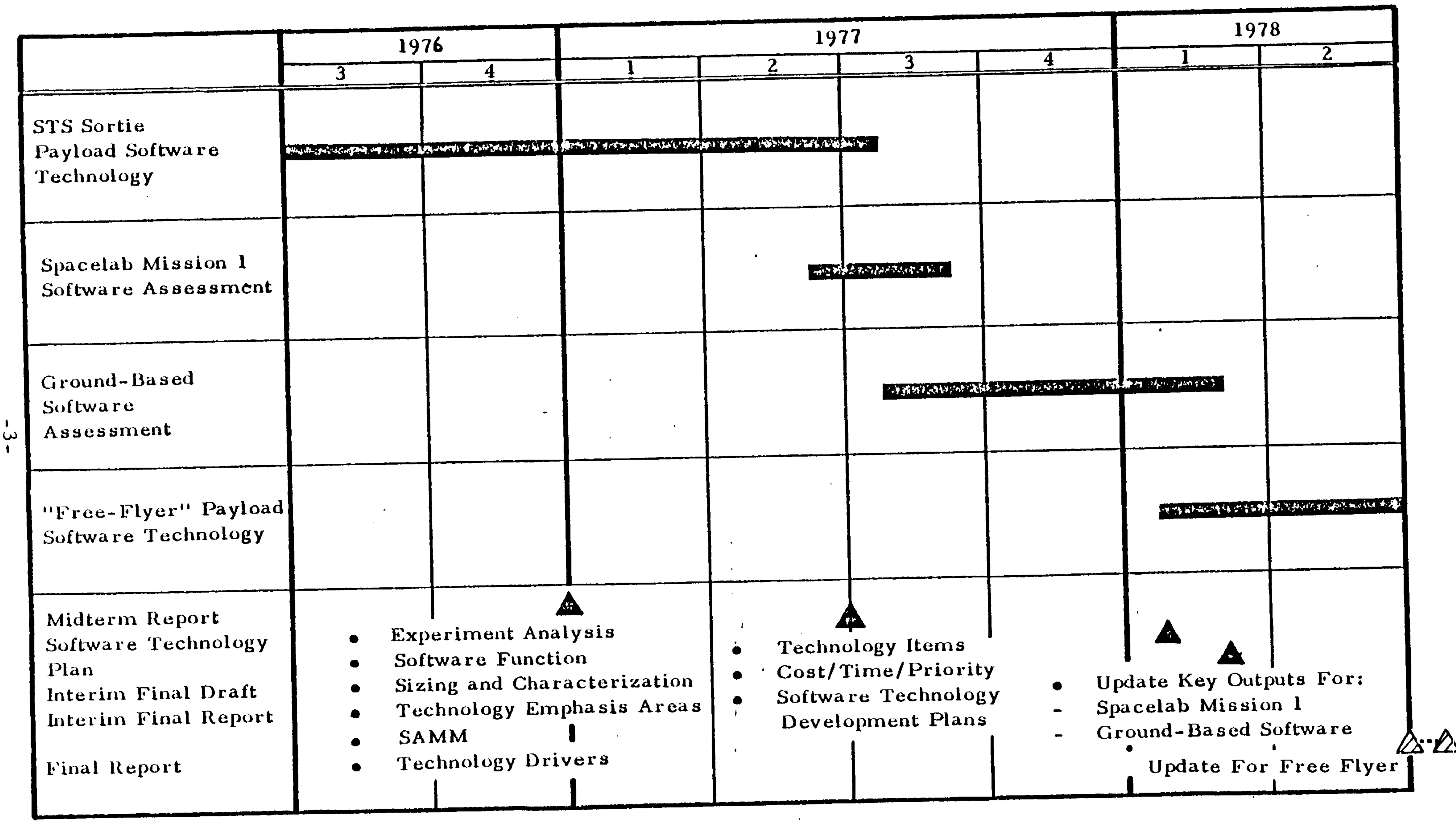


Figure 1-2

in the July 1975, Shuttle Sortie Payload Description (SSPDA), and from OFS and OAST space technology forecasts. This data was supplemented with an in-depth review of detailed payload studies, Design Reference Mission Analysis (DRM), Integrated Mission Analysis and Planning (IMAP), user's information, previous space programs and studies, and in-house expertise.

The technology development plans resulting from this study must be implementable during the 1980's in order to meet the requirements in the 1990's. It is assumed, for this study, that the primary development vehicle during this time will be the Space Transportation System. Therefore, payload analyses performed were limited to those payloads currently planned to be carried on the Space Transportation System.

Finally, it should be noted that this study was primarily concerned with software technology; not with computer systems technology. A specific effort has been made to limit the consideration of Technology Items to pure (or near-pure) software technology development. No attempt was made to analyze future Data Management Systems, as a whole, to drive out technology requirements. Such an effort certainly has merit, and might result in software technology requirements; however, many software technology requirements can be identified without an end-to-end data management analysis.

1.3 Organization

This document is organized into three sections as shown in Table 1-1. Section I is standard introductory material and is self-explanatory.

Section II is an executive summary providing an overview of the Payload Software study effort. It gives a brief summary of the software technology development studies contained in the plan, provides an overall schedule, and identifies critical sequential or complementary relationships. This section briefly summarizes related technology goals, either in hardware or software, being pursued elsewhere. It also lists the priority relationships so that the plan can easily be adjusted to varying budgetary conditions.

Section III provides a detailed study plan for each selected Software Technology Development Item. Each Software Technology Development Item is allocated a subsection which describes the conditions leading to a technology need, and relates the problem to supporting data accumulated during Phases I and II of the study. Each subsection defines potential cost and performance benefits and provides a summary of the anticipated development cost along with the apparent technological solutions and

PHASE III REPORT OUTLINE

SOFTWARE TECHNOLOGY DEVELOPMENT PLAN

SECTION I - INTRODUCTION - This section describes the purpose, scope, and organization of the Software Technology Development Plan.

SECTION II - SUMMARY - This section is an executive summary of the Software Technology Development Study.

SECTION III - DESCRIPTION OF SOFTWARE TECHNOLOGY ITEMS - This section contains the description and planned approach to the implementation of each selected Software Technology Development Item.

3.1 Item TI-01

.

.

.

3.28 Item TI-28

Table 1-1

associated technology development requirement(s). Associated hardware or system technology items are described to ensure that all facets of the problem, as well as its solution, are clarified. The potential approach that is likely to accomplish the stated technological objective is outlined. Schedule and resources required to accomplish each particular technological development item are identified. Means of intermediate and final measurable verification of the results are described, as applicable.

1.4 References and Bibliography

A thorough survey was made of current literature as well as government, industry, and other publications relating to software technology. The prime sources used in this study are as follows:

- Outlook for Space, NASA Document, January 1976.
- A Forecast of Space Technology 1980-2000, January 1976.
- OAST Space Theme Workshop, NASA Document, April 1976 (Quick-Look Comments and Working Papers).
- Space Electronics Technology Summary, March 1976.
- OAST Summer Workshop, NASA Document, August 1975.
- Information Processing/Data Automation Implications of Air Force Command and Control Requirements in the 1980's (CCIP's), 1975.
- Findings and Recommendations of the Joint Logistics Commanders Software Reliability Work Group, November 1975.
- "Key Developments in Computer Technology: A Survey," IEEE Computer Magazine, November 1976.
- IEEE Proceedings, Second International Conference on Software Engineering, IEEE Catalog No. 76CH1125-4C, October 1976.
- "New Directions in Space Electronics," Peter R. Kurzhals, Astronautics & Aeronautics Magazine, February 1977.
- Software Technology Present and Future, Bobby Hodges, 14th Annual Southeast Regional ACM Conference, April 1976.

2. SUMMARY

This section provides a top-level view of the Payload Software Technology Study, including the identification of technology development needs and an identification of development priorities. The Software Technology Items described in this section were selected in order to support efforts to:

- Reduce the high cost of meeting flight software reliability requirements.
- Remove limitations on information extraction techniques.
- Meet the requirements for human control of complex instruments and remote systems.
- Simplify, reduce the cost of and make usable complex software systems.

2.1 Payload Software Technology Study Overview

The four major tasks of this study (STS Sortie missions, Spacelab Mission I, supporting ground-based software systems, and Automated Payloads) were divided into three phases as shown in Figure 2-1.

Phase I was devoted to the evaluation of the processing requirements (string, speed, input/output) of representative payloads and provided the basis for selecting drivers of software technology. Phase I also included a review of technology forecasts to bound broad areas of need. Table 2-1 provides a list of technology emphasis areas which were derived primarily from the Outlook for Space study and OAST summer workshops. Consistent with expectations, drivers were not peculiar to specific missions, instruments, or applications; they were common to many classes of missions, instruments and applications.

The problems and needs thus identified were thoroughly assessed and compiled into a list of Software Technology Drivers as shown in Table 2-2. The Technology Drivers associated with Software Development must be considered of prime importance. Unless significant improvements are made in this area, all other related technologies will be stunted because the associated software will be too costly and/or too unreliable. In Software Systems Architecture, most of the Technology Drivers are related to LSI technology; i. e., the availability of very low-cost processing systems. These systems are particularly suitable for onboard usage and therefore receive significant emphasis. The remaining Technology Drivers are primarily based on anticipated increases in onboard processing of image type data, as well as on anticipated increases in the rates of data to be acquired.

PAYLOAD SOFTWARE TECHNOLOGY
STUDY PLAN

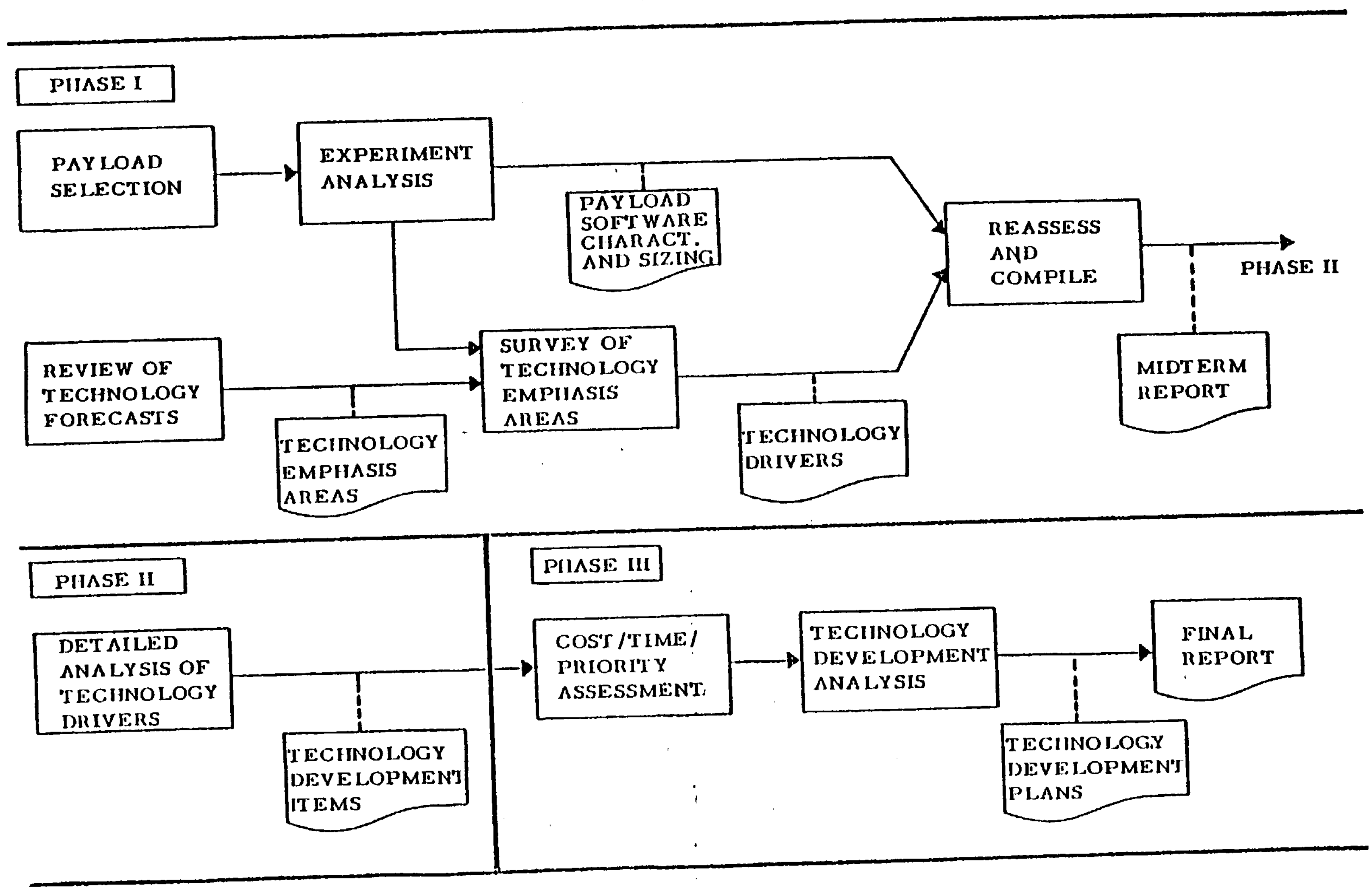


Figure 2-1

PAYLOAD SOFTWARE TECHNOLOGY
TECHNOLOGY EMPHASIS AREAS

- Software Development Technology.
  - Cost/Time Reduction Methods.
  - Software Reliability.
  - Cost/Performance Evaluation.
  - Software/Hardware Standardization.
- Software Systems Architecture.
  - Functional Distribution of Processing.
  - Fault Tolerant Systems.
  - Intelligent Instruments.
  - Human/System Interfacing.
  - Utilization of High-Rate Data Processors.
  - Data Distribution/Sharing Networks.
  - Very Large Data Base Management.
  - Multidimensional Data Base Systems.
- Software Application Technology.
  - Image Recognition Processing.
  - Data Compression.
  - Automated Intelligence.
  - Automation of Ground Support Functions.

Table 2-1

TECHNOLOGY DRIVER SUMMARY

- Software Development.
  - PS-01 Software Design Engineering.
  - PS-02 Trend Toward Software Development by Non-Programmers.
  - PS-03 Fault-Free Software.
  - PS-04 Application Oriented Language Design Methodology.
  - PS-05 Low-Cost Development of AOL Compilers.

- Software Systems Architecture.
  - PS-06 (Distributed) System Partitioning/Interconnection Techniques.
  - PS-07 Very Large Storage Access Simplification.
  - PS-08 Software Fault (Own or Induced) Detection.
  - PS-09 Software Recovery (After Fault Detection).
  - PS-10 High-Speed Buffering Techniques.
  - PS-11 Design and Control of Adaptive Software Procedures.
  - PS-12 Use of "Natural" Communication Methods.
  - PS-13 Efficient Large Array Search and Sort Procedures.
  - PS-14 Parallel Processing Techniques.
  - PS-15 Efficient Large Array Manipulation Procedures.
  - PS-16 Spacelab Mission I.
  - PS-17 More Effective Distribution of Processing.
  - PS-18 Greater Data Utility and User Participation.
  - PS-19 Efficient Storage and Retrieval of Remotely Sensed Data.
  - PS-20 Scene Analysis.
  - PS-21 Surface Navigation and Mapping.

Table 2-2

In Phase II of the study, the Software Technology Drivers were analyzed in detail in search of potential solutions to the problems posed. This analysis resulted in identification of many Software Technology Items. A number of these were determined to require forced advancement by NASA if current NASA plans are to be met. The Software Technology Items are shown in Figure 2-2 in relation to the Technology Drivers from which they were derived. (Blocks containing asterisks represent relationships established during Technology Development Analysis activities of Phase III. Asterisks in the TI-19 row indicate secondary benefits that may be derived from use of software prototyping.)

The total number of Technology Items identified may be impossible to implement within the resource constraints. Therefore, during Phase III, cost estimates and priorities were assigned to the Technology Items. The result is a recommended set of Software Technology Development Plans.

2.2 Relation to Other Technology Development

Dependencies exist between variable independent Technology Items. Solution of a problem posed by a Technology Driver requires the full consideration of the interrelationships between these variables as well as the direct relationship to a specific item.

Other software technology development activities are either underway or have been proposed, both within and outside NASA. The more significant of these are discussed in the Software Technology Item Derivation Worksheets (Volume I, Appendix G).

TECHNOLOGY DRIVER-ITEM CORRELATION TABLE

TI NO.	SOFTWARE TECHNOLOGY ITEM	SOFTWARE TECHNOLOGY DRIVER: PS- 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1	REQUIREMENTS DECOMPOSITION AND STRUCTURING GUIDELINES	X					X								X						X	
2	REQUIREMENTS-TO-CODE TRANSLATION AIDS		X																			
3	SIMPLIFIED SOFTWARE DEVELOPMENT SYSTEM DEMONSTRATION MODEL		X																			
4	QUERY-GUIDED IMPLEMENTATION METHODS		X																			
5	STANDARDIZATION CRITERIA FOR NASA SOFTWARE			X													X		X			
6	AOL DESIGN GUIDELINES/STANDARDS				X	X																
7	AOL COMPILER GENERATOR COST FACTORS					X																
8	INTERFACE CRITERIA FOR NASA DISTRIBUTED PROCESSOR APPLICATIONS	X					X								X		X	X			X	X
9	TASK CONTROL STRUCTURES FOR DISTRIBUTED PROCESSOR ENVIRONMENTS						X								X						X	X
10	PROGRAM ORGANIZATION METHODS FOR REAL-TIME FAULT RECOVERY						X			X					X							X
11	ADAPTIVE HIGH SPEED BUFFERING TECHNIQUES FOR DYNAMIC NETWORKS										X	X							X			
12	CONTROL STRUCTURES FOR ADAPTIVE SYSTEMS											X										X
13	IMPACTS OF NATURAL COMMUNICATION METHODS ON NASA PAYLOAD SYSTEMS												X				X					
14	ADAPTIVE SEARCH AND SORT PROCEDURES											X		X						X	X	
15	RESTRUCTURED IMAGE ANALYSIS SOFTWARE FOR PARALLEL PROCESSING														X						X	
16	OPTIMAL LARGE ARRAY PARTITIONING PROCEDURES															X				X		
17	CLASSIFICATION OF ERROR MECHANISMS		X	X					X	X												
18	GUIDELINES FOR DESIGN DOCUMENTATION CONSISTENCY	X			X																	
19	SOFTWARE PROTOTYPING METHODS	X	*	*			*	*	*	*	*	*		*	*	*	X	*	X			
20	SOFTWARE TECHNOLOGY MONITORING	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
21	SPACELAB SOFTWARE TECHNOLOGY EXPERIMENT	X	X		X								X				X					
22	ONBOARD RADIOMETRIC CORRECTION						X				X							X				
23	ONBOARD CLASSIFICATION AND DISTRIBUTION										X				X				X			
24	SECURITY AND CONTROL OF HIGH-RESOLUTION DATA											X							X			
25	LOW-COST USER SYSTEMS				X	X													X			
26	EARTH MODEL DATA BANK							X						X		X				X		
27	SCENE CONTENT EXTRACTION											X		X		X					X	
28	PATH PLANNING TECHNIQUES											X		X		X						X

Figure 2-2

2.3 Identification of Priorities

Many factors were involved in setting the priorities of the candidate Software Technology Items. Once the initial step of identifying a problem was completed, the next step was to evaluate the current and estimated state of the art in the associated technology area. This evaluation is reflected in the Cost/Time/Priority Assessment. The levels established were based on the following criteria:

1 - Basic principles observed and reported.

2 - Conceptual design formulated.

3 - Conceptual design tested analytically or experimentally.

4 - Critical function/characteristic demonstrated.

5 - Component/section tested/reviewed in relevant environment.

6 - Prototype/draft tested/reviewed in relevant environment.

7 - Final product tested in operational/space environment.

8 - New capability derived from a much lesser operational model.

9 - Reliability upgrading of an operational model.

These state-of-the-art levels were not used for setting priorities per se, but were used as aids to help determine whether appropriate advancement could be achieved by industry alone, or if NASA participation would be required.

In order to facilitate setting priorities, attain consistency, and limit subjectivity, a logic flow was developed as shown in Figure 2-3. This flow embodied the major elements of priority assignment and resulted in a rating of 0-6 for each technology item. The elements used were:

- Enablement.
- Cost Reduction.
- Enhancement.

Enablement was subdivided into two categories, critical and desirable. (Initially, enablement was divided into mandatory and desirable, but as it became evident that few of the items would actually preclude a space mission, the mandatory rating was downgraded to critical).

TECHNOLOGY ITEM PRIORITY

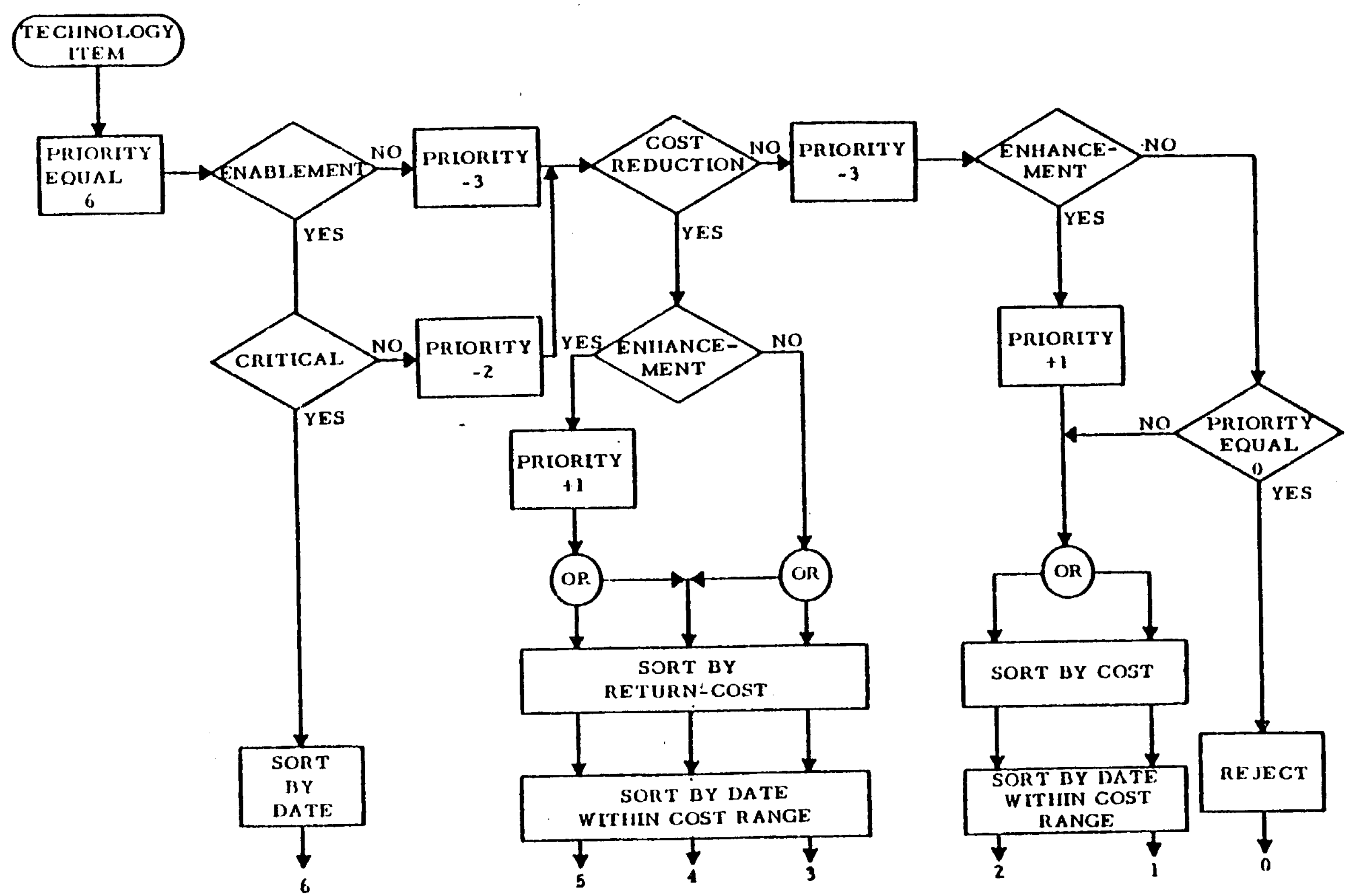


Figure 2-3

Critical enablement describes an item which could gravely affect the achievement of a major NASA objective (e.g., 1000x data at 1/10th cost) if it were not resolved. Critical enablement was given a weight of 6. Desirable enablement describes a technology that would enable a new function to be performed, or allow additional experimentation with a given instrument, that would significantly increase the return benefit of a payload. Desirable enablement was equated in value to enhancement, and thus given a weight of 1.

Cost reduction applies to those software items which could have major effect on the cost of space activity in general. Most of these items fall in the category of Software Engineering or Software Development. Cost reduction was given a weight of 3.

Enhancement applies to those items that would enhance the total information gathering and distribution process. This includes improved reliability, more effective processing, higher quality data recovery, and other advancements that would be highly beneficial. Enhancement was given a weight of 1.

After the initial assessment, other factors (Drivers affected and Cost/Benefit Returns) were added in. A value of 1 was added to each Technology Item for every Driver upon which it had a primary effect. The Cost/Benefit Return was given a rating of high, medium or low, and equated to a weight of 2 for high, 1 for medium and 0 for low. (Note: Need dates were determined not to be significant factors for these Technology Items.)

The priority factors were then summed for each Software Technology Item to derive an overall rating as shown in the priority evaluation chart in Figure 2-3. The Technology Items identified with an asterisk are those that fall in the top third of the overall ratings, and thus are those recommended for immediate NASA consideration.

2.4 Development Schedule

A schedule for the development or advancement of technology required by each of the selected Software Technology Items is contained within each Technology Development Plan. In general, each item is given a 1978 or 1979 start date, as most of them are immediately necessary with the advent of the STS operational phase in 1980 and each will require a year or more to complete.

For planning purposes it was assumed that development of an operational application would take two years after demonstration of a

PRIORITY EVALUATION

TI No.	Critical Enablement	Desirable Enablement	Cost Reduction	Enhancement	Priority	"Free-flyer" Impact	Spacelab Mission I Impact	Applicable Drivers	Cost/ Benefit Return	Need Date	Rating	Top 1/3
01		X	X	X	5	2	1	4	1	N/A	13	X
02			X		3	2	0	1	0	N/A	6	
03			X		3	0	0	1	1	N/A	5	
04			X		3	0	0	1	0	N/A	4	
05			X	X	4	0	1	3	2	N/A	10	
06			X		3	0	0	2	0	N/A	5	
07			X		3	0	0	1	0	N/A	4	
08		X	X	X	5	2	1	7	2	N/A	17	X
09		X	X	X	5	2	1	4	1	N/A	13	X
10		X	X	X	5	2	0	4	1	N/A	12	X
11		X		X	2	1	0	3	0	N/A	6	
12		X			1	2	0	2	0	N/A	5	
13				X	1	0	0	2	0	N/A	3	
14		X			1	2	0	4	0	N/A	7	
15		X	X	X	5	2	0	2	2	N/A	11	X
16		X			1	2	0	2	0	N/A	5	
17			X	X	4	2	0	4	0	N/A	10	
18			X		3	1	0	2	0	N/A	6	
19			X	X	4	1	1	3	1	N/A	10	
20		X	X	X	5	0	0	21	2	N/A	28	X
21		X	X	X	5	0	0	5	2	1980	12	X
22		X	X		4	1	0	3	1	N/A	9	
23		X	X	X	5	0	0	3	2	N/A	10	
24		X			1	1	0	2	0	1980	4	
25			X	X	4	0	0	3	2	N/A	9	
26		X	X	X	5	0	0	4	2.	N/A	11	X
27	X				6	0	0	4	2	N/A	12	X
28		X		X	2	0	0	4	2	N/A	8	

Figure 2-4

prototype model. For instance, an image processing application needed in 1985 would have to be proven feasible by 1983 in order to allow time for development of a flight item.

In any event, the benefits that will accrue from the development of any Technology Item cannot begin until the development is complete.

3. SOFTWARE TECHNOLOGY ITEM DEVELOPMENT PLANS

3.1 Requirements Decomposition and Structuring Guidelines (TI-01)

3.1.1 Problem Definition

Software design practice up to now has suffered from a lack of consistency among designers in their approach to decomposition of the requirements and development of design structure. The need for rational guidelines becomes apparent when one considers the fact that totally different designs can result from different designers working on identical software requirements. This makes it almost impossible to evaluate designs, much less to identify the optimum design. Unlike hardware, in which standard components are integrated by well-established logic principles, each software design is a new design. This is costly in development and implementation, but even more so in the training and maintenance requirements to support utilization phases of an operational system.

Of even more importance, lack of consistent, good design compromises the most valuable characteristic of software -- the flexibility to adapt to changing requirements. The study proposed to help solve this basic Software Engineering problem is expected to be of moderate cost (3-10 man years).

3.1.2 Technology Development Requirements

The technology developments needed under this broadbased study are the guidelines for requirements decomposition and structuring in software design. These developments must contribute towards achieving consistency in design. Also, it is necessary to derive and establish proper design evaluation criteria.

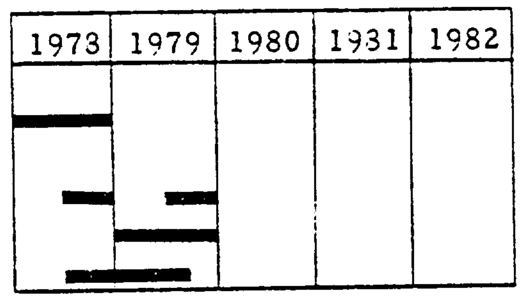
Technology must be developed which will provide a means to convey what the requirements are, and will allow the breakdown of these requirements into a functional distribution. This should lead to software designs that are reliable and easily understood, implemented, used, and maintained.

3.1.3 Suggested Approach

The study proposed for this effort has four steps as shown under the schedule. The first step is the establishment of requirements decomposition guidelines. A comprehensive set of guidelines for decomposition of the furnished software requirements is to be developed. This will be done through a thorough review of the various current approaches to the problem and a critical analysis of the relative advantages and disadvantages of these approaches. The next step is to develop a logical set of guidelines aimed at achieving consistency of design across a given set of requirements. This must be followed by the establishment of techniques which will ensure proper structuring of functions into a viable, usable design. Development of guides to proper structure and consistent design is an iterative enhancement process, as shown in the schedule. In parallel with the previous activity, the criteria for measurement and evaluation of design products will be established. This will be of considerable significance in assessing the results of using these guidelines in NASA applications.

3.1.4 Schedule and Resources

- o Requirements Decomposition Guidelines.
- o Design Consistency Guidelines.
- o Structuring Guidelines.
- o Evaluation Criteria.



Estimated Cost of Development: $150K - 500K.

3.1.5 Verification of Results

The end product of this effort will be a set of guidelines for software design rather than a specific piece of software. The verification of results can be viewed as the assessment of benefits arising from application of these guidelines. Thus, this step in effect consists of applying these guidelines to one or more specific software design efforts to compare the experience with similar efforts not based on these guidelines. The relative merit will be assessed through application of the evaluation criteria, which in turn can be appraised through demonstration of actual program product results. The verification process itself will make a valuable contribution to the development of the Software Engineering profession.

3.1.6 Related Studies

- o Software Technology Items.
  - TI-08 Interface criteria for NASA distributed processor applications.
  - TI-09 Task control structures for distributed processor environments.

- TI-18 Guidelines for design documentation consistency.

- TI-19 Software prototyping methods.

o Software Technology Items (Secondary).

- TI-10 Program organization methods for real-time fault recovery.

- TI-15 Restructured image analysis software for parallel processing.

3.2 Requirements-to-Code Translation Aids (TI-02)

3.2.1 Problem Definition

Current practice of manual generation of codes from given software requirements is not only expensive but also error-prone. This is because, first, the software requirements are not specified in a standard format and hence the requirements decomposition is mostly adhoc. Second, the lack of consistent design guidelines makes the steps leading to the generation of codes more of an art (or lack of it!) rather than a science. Therefore, for a given set of requirements the generated code does not necessarily reflect the best code, and methods of proving the software as fault-free are still to be realized. Therefore, some means of automating the code generation process is essential to achieve a measure of error and cost control in software development. This requirement-to-code translation problem is likely to be of relatively high cost and require 10-20 man years.

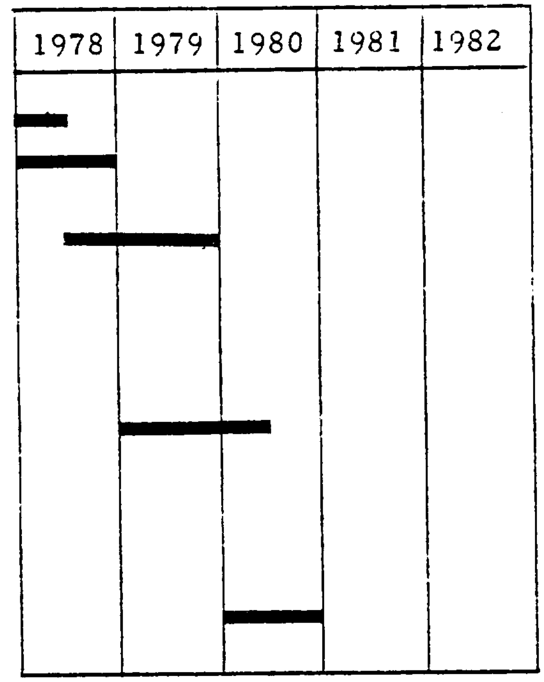
3.2.2 Technology Development Requirements

The requirements under this study are defined as the development of automated aids for generating program codes (in a desired language) from a specified set of software requirements. This technology development is tailored to meet the specific needs of payload software requirements, rather than any general study of automatic programming, to ensure direct payoff for NASA for its investment in this technology development effort.

3.2.3 Suggested Approach

The first step of this study is a review and assessment of the on-going efforts in this field as applied to other areas of software development, particularly business-oriented and other commercially funded projects. In parallel with this, a requirements analysis must be carried out to identify the spectrum of NASA payload software needs. With these preliminary efforts as the base, a master software instructions/code library, consisting of all the key functions reflecting NASA payload software requirements, can be developed. The next step is to define and develop a transformational approach which can successively transform, in stages, the input software requirements into a set of program codes using the master library as a reference. The developed methodology is then tested and the related documentation prepared to ensure its suitability for application to NASA payload software development.

3.2.4 Schedule and Resources



- o Analyze requirements.
- o Identify the spectrum of NASA payload software requirements.
- o Develop a master software instructions code library consisting of all basic functions reflecting NASA payload software requirements.
- o Develop a transformational approach which transforms the input requirements successively into a set of program codes using the library.
- o Test and document the methodology developed.

Estimated cost of development: \$500K - \$1000K.
Resources include access to information on future payload software development plans and program details.

3.2.5 Verification of Results

The developed requirement-to-code translation aids will be test-implemented and validated as to their ability to perform the designated function. The testing will include actual application of the code generator product to a previous software implementation to derive and compare the operational capability of the automatically generated code with the earlier manually developed code.

3.2.6 Related Studies

- o Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-03 Simplified software development system demonstration model.
  - TI-04 Query-guided implementation methods.
  - TI-18 Classification of error mechanisms.
  - TI-19 Software prototyping methods.

3.3 Simplified Development System Demonstration Model (TI-03)

3.3.1 Problem Definition

The advent of large scale usage of microprocessors in distributed systems for real-time space applications calls for simpler software development tools. This is necessary to ensure that the cost of software development will not become a barrier to the benefits of microprocessor usage which otherwise is very cost-effective.

Many software development systems exist, but it is rare to find one that is simple and even rarer to find one that is transportable. In an era of diverse users of space systems, each developing a software subset of a future integrated flight complement, ways and means of simply and economically developing software components must be found.

An effort which is designed to achieve this objective is the creation of a demonstration model of a simplified system for software development. This effort can be expected to be one of relatively high cost requiring 10-20 man years.

3.3.2 Technology Development Requirements

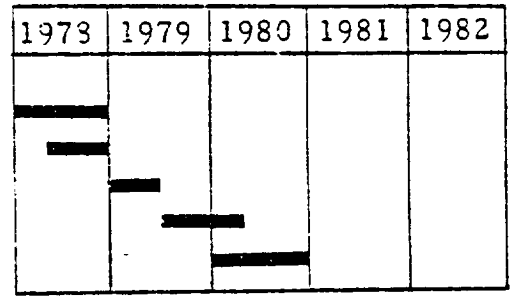
The feasibility of a simplified system for software development can be shown by the creation of an appropriate demonstration model. Accordingly, the technology requirements of the study can be stated as the building of a simplified development system demonstration model. This will be a precursor to low-cost operational systems for use by non-professional programmers in the development of space applications.

3.3.3 Suggested Approach

This development effort has several identifiable steps as shown in the schedule. The first step is the requirements definition. Requirements definition is based on developing a clear understanding of the purpose of the model and the environmental constraints under which the modeled system will have to operate. These requirements are analyzed to derive the specifications of the demonstration model. This is followed by the design of the demonstration model which will emphasize the different aspects of the operational system envisioned for the PI's use. The design phase leads to the development and test phase which in turn takes the project to the final detailed system evaluation phase. The final evaluation phase is a key element. It must be detailed enough to ensure that the follow-on work of developing an operational system can be taken up with confidence that the end product will support the anticipated heavy schedule of software development in the coming decade.

3.3.4 Schedule and Resources

- o Requirements Definition.
- o Requirements Analysis
- o Demonstration Model System Design.
- o Development and Test.
- o Evaluation.



Estimated cost of development: $500K - $1000K.

3.3.5 Verification of Results

The evaluation of the demonstration model and its effectiveness is a significant part of the effort to be accomplished under this study. The schedule allows a full year for this activity and effectively portrays the importance given to this phase of the effort. This is essential to ensure the integrity of the demonstration model and the operational system to follow. Each step of the effort provides concrete output products (i.e., requirements, composition, design specifications, test system and evaluation system) for measurable verification of progress.

3.3.6 Related Studies

- o Software Technology Items.
  - TI-02 Requirements-to-code translation aids.
  - TI-04 Query-guided implementation methods.
  - TI-17 Classification of error mechanisms.
  - TI-19 Software prototyping methods.

3.4 Query-Guided Implementation Methods (TI-04)

3.4.1 Problem Definition

The objective of this technology item is the development of new programming procedures wherein the programming effort is supported/ guided by a questionnaire. The basic problem to be addressed is essentially similar to the one underlying TI-02, Requirements-to-Code Translation Aids, in that new automated tools for software development are called for to reduce the expensive, error prone manual approach to code generation. However, the emphasis here is less on autonomy than in the other approach and presumably the objectives are more easily realizable. This effort is consequently more moderate in cost (5-10 man years) and is expected to result in a faster return of benefits.

3.4.2 Technology Development Requirements

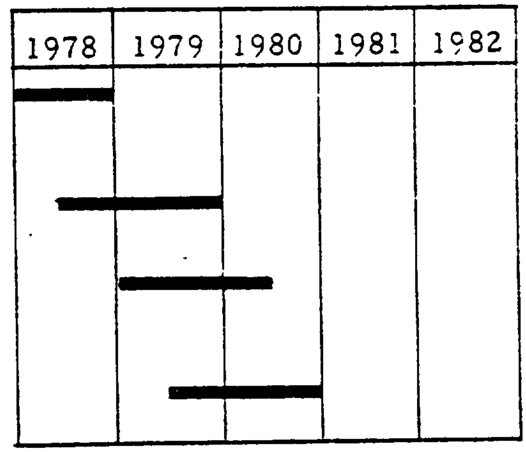
Technology development required under this study are semi-automated tools for software development using the questionnaire approach. This questionnaire approach must be especially attractive for interactive man-machine systems, operating in dynamic environments, wherein it may be necessary to create new software and/or new configurations of existing software. Query-guided implementation methods must offer the flexibility necessary for such applications. At the same time these methods must contribute towards the traditional software development goals in terms of reduced software errors and consistency in the quality of the resulting software product.

3.4.3 Suggested Approach

As a practical step, it is advisable to constrain this effort of the development of semi-automated tools to the specific NASA payload software development requirements. This limitation ensures that a viable solution will be developed within the time and cost constraints of NASA programs. Any attempt at global solutions could lead to lack of effective control over the projected costs and time schedules. The first step is, therefore, to develop and establish the common features and requirements of NASA payload software development programs. This is to be followed by an effort directed towards developing a skeleton structure of the software corresponding to established commonalities. The next step is the design and development of a questionnaire methodology which when applied to specific requirements leads to the filling out of the skeleton structure and results in the desired software package. Finally, the methodology should be thoroughly tested and documented to ensure an appropriate return on investment for NASA.

3.4.4 Schedules and Resources

- o Derive and establish the common features of NASA payload programs.
- o Develop a skeleton structure of the development system.
- o Develop a questionnaire methodology for filling out the skeleton structure.
- o Test and document the methodology.



Estimated development costs: \$200K - \$500K

3.4.5 Verification of Results

The end product of the study being implementable methodology, it is necessary to ensure that the developed tools can be put to effective use through testing and documentation. This testing should include test implementation in specific NASA software development programs to evaluate its effects on efficiency, reliability and cost of program development. In view of the man-machine interaction involved in questionnaire approaches, it is desirable to study programmer responses to the methodology as part of the verification process to ensure its acceptability.

3.4.6 Related Studies

- o Software Technology Items.
  - TI-02 Requirements-to-code translation aids.
  - TI-03 Simplified software development system demonstration mode.
  - TI-18 Classification of error mechanism.
  - TI-19 Software prototyping methods.

3.5 Standardization Criteria for NASA Software (TI-05)

3.5.1 Problem Definition

Lack of standardization in software design, development, testing and other facets of software projects, even within the bounded sphere of NASA activities, frequently results in expensive, often duplicated software. Needed reliability is difficult to achieve. Standardized software can make a significant contribution towards the goal of totally fault-free software. Therefore, as a first step, it is necessary to develop the criteria for standardization. The degree of the problem solution depends on the scope of the standardization process; however, as proposed here is expected to be of moderate cost involving 3-10 man years of effort.

3.5.2 Technology Development Requirements

Development of standardization criteria is the goal of this technology development item. This criteria development will span the spectrum of activities associated with a software development project. For example, standardization has many implications: standard language design, standard operating system design, and standard interface implementation. A degree of standardization in each of these areas and criteria for such standardization is desired and sought under this effort.

3.5.3 Suggested Approach

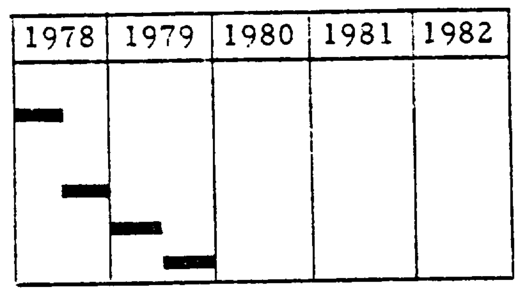
The first step must be to understand and analyze the implications of standardization. Through this effort, it should be possible to identify the specific elements of a software project which can be considered prime candidates for standardization. Once the areas deemed feasible for standardization are established, the next logical step is to develop the criteria for standardization.

It is advisable to gain some insight into industry practices in standardization and apply these ideas with necessary modifications to the NASA environment. Modification and possibly new concepts may be necessary in view of (1) the decentralized nature of the NASA organization (and the diversified contractor force supporting it), (2) the consequent problems of coordinating the software development efforts, and (3) the difficulties in enforcing the standardization procedures. Selected candidate software packages should be weighed and tested against the developed standardization criteria to validate their effectiveness in establishing standard NASA software libraries for the 1980-1990 time frames.

New developments in software design methodology which might influence the standardization criteria are conceivable and modifications at a later date may have to be made. However, irrespective of such future changes, it is necessary to have fairly rigid standardization criteria to avoid duplication of efforts and minimize the software development costs and associated errors.

3.5.4 Schedule and Resources

- o Analysis of Standardization Implications.
- o Development of Criteria.
- o Identification of Candidates.
- o Evaluation of Validity.



Estimated cost of development: $150K - $500K.
Resource requirements include NASA-wide access to software projects, programs and present practices for minimizing duplication of efforts.

3.5.5 Verification of Results

The end product of this study is a set of standardization criteria. The verification process essentially consists of applying the criteria to selected software packages and development projects to determine the validity and feasibility of enforcing such criteria in the development process.

3.5.6 Related Studies

This Technology Item is pertinent to virtually every aspect of the computer software domain.

3.6 AOL Design Guidelines/Standards (TI-06)

3.6.1 Problem Definition

Application-oriented languages (AOL) are expected to dominate the scene in future decades with users developing the software required for their particular needs. This requires appropriate AOL design guidelines and standards to ensure that the AOL's developed to meet the needs of a changing environment are compatible with NASA system goals and have the requisite language consistency. This developmental effort is expected to be one of relatively modest cost involving 2-5 man years.

3.6.2 Technology Development Requirements

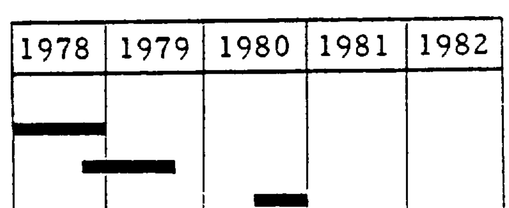
Development of design guidelines and standards for application-oriented languages (of interest to NASA applications) is the technology advancement required by this item. This requirement is the first step towards the goals of economical development of efficient application-oriented languages, particularly those of real-time onboard distributed systems. Design guidelines/standards are to ensure that the resulting AOL's are simple to use and efficient even when deployed by non-professional programmers. These requirements tie this effort to the one proposed in the next item (TI-07) on AOL compiler generator cost factors. The AOL design guidelines will determine the complexity of the compilers and therefore influence the cost of generating these compilers. Accordingly, the developments under this study have in view the objectives and associated factors of TI-07.

3.6.3 Suggested Approach

As illustrated in the schedule, the approach consists of three specific steps. The first is the determination of the concepts underlying AOL and understanding what its requirements are. These concepts are an essential prerequisite to the development of guidelines for AOL design. The next step is the actual development of guidelines to meet the requirements as derived. The third step is the formulation and development of criteria to evaluate the AOL design resulting from the use of these guidelines. This provides a means of verifying the applicability and usefulness of the derived guidelines in practical problems of AOL design.

3.6.4 Schedule and Resources

- o AOL Requirements Concepts.
- o AOL Design Guidelines/Standards.
- o Design Evaluation Criteria.



Estimated cost of development: $100K - $250K.

3.6.5 Verification of Results

The guidelines derived for the study must be applied to specific AOL design projects to test the applicability of these guidelines to real-world NASA problems. The evaluation criteria will then be deployed as the tool for assessing the resulting design, which reflects on the usefulness of the derived guidelines in providing an effective AOL to an actual user.

3.6.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-07 AOL compiler generator cost factors.
  - TI-18 Guidelines for design documentation consistency.

3.7 AOL Compiler Generator Cost Factors (TI-07)

3.7.1 Problem Definition

The advent of increased development and usage of application-oriented languages (AOL) calls for design of an increasing number of compilers. Compilers being machine-dependent, the problem is further enlarged by the continuing spread in the number of different computers coming into usage. Compiler generation, however, has remained an expensive task. Thus in view of this expected increase in compiler design activities, it is necessary to look into why compilers cost so much. This study of investigating the factors contributing to the cost of compiler generation is viewed as one of relatively modest effort (2-3 man years). This will lead to methods of significantly reducing compiler generation cost.

3.7.2 Technology Development Requirements

The requirements of this task are of a different nature from the other technology items. The end product is not a specific technique or software package, but a qualitative assessment of the major factors contributing to the cost of compiler design, from which it should be possible to derive techniques for reducing these costs.

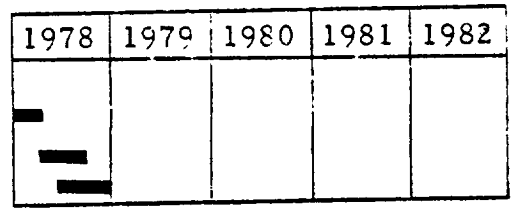
That problems exist in compiler generation is a known fact. Why they exist is another question. The requirement of this task is to provide an answer in quantifiable terms that will lead to new and feasible approaches to the development of tools to support application growth.

3.7.3 Suggested Approach

The study to be undertaken consists of three steps as shown in the schedule. The first step is to identify and typify representative AOL compiler generators used to generate payload application software. This will be followed by an in-depth analysis of the development history of these generators. The analysis must take into account the particular environmental constraints and other factors associated with the development of each of these generators. The analysis must identify the development cost and extract those factors contributing to the cost. Each of the contributing factors must be assessed in terms of significance and potential impact on future development programs. The result will be a comprehensive overall assessment of the problem of controlling costs in AOL compiler generation, and the identification of what steps must be taken to reduce future cost. This effort should be correlated with that under TI-06 to ensure that the final assessment reflects the benefits of development under TI-06.

3.7.4 Schedule and Resources

- Identify typical AOL generators.
- Trace development history.
- Perform cost assessment.



Estimated cost of development: $100K - $150K.

3.7.5 Verification of Results

The end product will be a specific set of factors which will provide a tangible product in terms of the benefits accrued in future compiler generation programs. As such the test would be to apply the experience gained by the analysis of these cost factors in a test compiler generation exercise and to assess the associated costs relative to the costs of prior comparable programs.

3.7.6 Related Studies

- Software Technology Item.
  - TI-06 AOL design guidelines/standards.

3.8 Interface Criteria for NASA-Distributed Processor Applications (TI-08)

3.8.1 Problem Definition

The expanding role of distributed processor systems envisioned for future payloads and missions demands that the associated problems of interprocess communication and related aspects be studied in depth by NASA. This problem is being addressed for onground systems within the industry. However, the special implications of onboard near-real-time processing environments need to be carefully examined by NASA. The development of suitable interface criteria for NASA-distributed processor applications is one of modest cost (2-4 man years). However, it is mandatory in terms of integrating and absorbing the multiplicity of high rate data sources projected for the next decade.

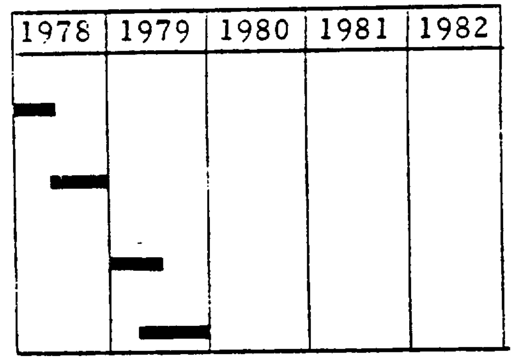
3.8.2 Technology Development Requirements

Interface criteria to assist in the design of NASA-distributed processor interfaces represent the major end products expected of this technology development effort. The criteria are necessary to simplify problems of integration and to minimize associated costs by avoiding recourse to nonstandard interfaces. These developments are critical to increased and cost-effective utilization of distributed processor concepts and techniques, and are essential to the enablement of intelligent instruments.

3.8.3 Suggested Approach

The first significant step of this effort is the identification of the potential distributed system configurations for deployment in near-real-time spaceborne environments. Once these are identified, it is necessary to analyze in detail each of these configurations to determine parametric characteristics such as data flow, and to establish appropriate control structures. These parameters characterize the interprocess communication needs. Based on these quantitatively established needs, the appropriate interprocess communication techniques can be selected through trade studies. This will in turn lead to the interface criteria which are essential to widespread deployment of distributed processor systems within the NASA programs.

3.8.4 Schedule and Resources



- o Identify potential distributed configurations.
- o Characterize data and control flow.
- o Select interprocess communication techniques.
- o Establish interface criteria.

Estimated cost of developments: $100K - $200K.

3.8.5 Verification of Results

The end product is a set of interface criteria. The test of its validity lies in its application to real-world problems, and verification can be achieved through test implementation of selected cases.

3.8.6 Related Studies

o Software Technology Items.

- TI-01 Requirements decomposition and structuring guidelines.
- TI-05 Standardization criteria for NASA software.
- TI-09 Task control structures for distributed processor environments.
- TI-10 Program organization methods for real-time fault recovery.
- TI-15 Restructured image analysis software for parallel processing.

3.9 Task Control Structures for Distributed Processor Environments (TI-09)

3.9.1 Problem Definition

Task allocation and control is major significance in the design and operation of ground-based distributed systems and has received considerable attention. This function is much more critical in the management of real-time space-based distributed systems and it is decidedly in the NASA domain to tackle this problem. The task allocation process should ensure that the available resources are utilized to the fullest extent at all times and as efficiently as possible, given the particular tasks and processor configuration. Once the optimal task allocation is determined, then the actual task control should be implemented in accordance with the planned allocation. Development of the task control structure represents the other major part of the problem contained in this item. It is expected that, in view of the progress in ground-based systems, this effort will be of relatively low cost and require 2-5 man years of effort.

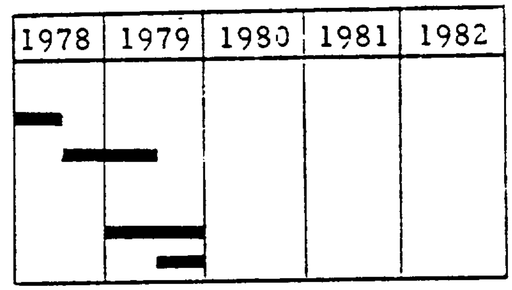
3.9.2 Technology Development Requirements

The technology development requirements are twofold: (1) techniques for automatically determining the optimal task allocation and scheduling among the processors within the distributed system, and (2) design and development of control structures for enforcing the optimal task allocation and scheduling. These tasks are interrelated and should be carried out with close coordination between the two efforts. This item has particularly great applicability to the effort proposed under TI-15 (restructuring image analysis software for a parallel processing). The technology development efforts under this plan should therefore complement the other effort.

3.9.3 Suggested Approach

The starting point for this plan is to clearly identify the scope and type of tasks as well as the nature of the distributed system environments expected to arise in the implementation of planned NASA missions. Concurrently, available techniques for onground distributed systems should be reviewed to delineate the deficiencies of such methods in meeting the requirements of real-time space systems. The two efforts together will provide the technical base for initiating the development of appropriate task allocation and scheduling techniques. These techniques should be amenable to automated implementation. This will lead to optimal allocation and scheduling of the tasks for any given processing problem within the scope of the study. Next will be the development of suitable software control structures capable of implementing the optimal allocation and scheduling plans. The final step is testing of these techniques and control structures, through simulation of the onboard processing environments, to verify their reliability and efficiency.

3.9.4 Schedule and Resources



- o Requirements analysis.
- o Develop optimal task allocation and scheduling methods.
- o Develop task control structures.
- o Test and documentation.

Estimated cost of development: $100K - $250K.
Resource requirements include computer facilities for simulation and testing.

3.9.5 Verification of Results

The results in this case are the techniques for determining the optimal task allocation and scheduling and the control structures for their implementation. The verification process should therefore include simulation of the onboard real-time processing environment and testing of the developed techniques and control structures in such simulated environments. This will ensure that the resultant products of the study will meet the NASA requirements for distributed systems. It will also enable operational implementation of related technology developments (particularly the implementation of parallel processable image analysis functions: TI-15).

3.9.6 Related Studies

- o Software Technology Items.
    - TI-01 Decomposition and structuring guidelines.
    - TI-05 Standardization criteria for NASA software.
    - TI-08 Interface criteria for NASA distributed processor applications.
    - TI-10 Program organization methods for real time fault recovery.

- TI-15 Restructured image processing software for parallel processing.

- TI-19 Software prototyping methods.

3.10 Program Organization Methods for Real-Time Fault Recovery (TI-10)

3.10.1 Problem Definition

Software and hardware errors as well as data anomalies do occur in spite of the best of precautions in the design and development processes. It is therefore necessary that space systems should be backed by effective means of fault recovery. In view of the dynamic nature of the system environment, it is also crucial that this recovery be in near-real-time. Thus, the problem addressed in this study is the need for developing efficient techniques for real-time recovery from errors caused either by hardware, data, or software. The recovery procedures to be considered here are primarily limited to software rather than to purely hardware-oriented solutions such as redundant hardware. The proposed effort is of moderate to high cost (between 4-20 man years) depending on the depth to which the problem is to be explored.

3.10.2 Technology Development Requirements

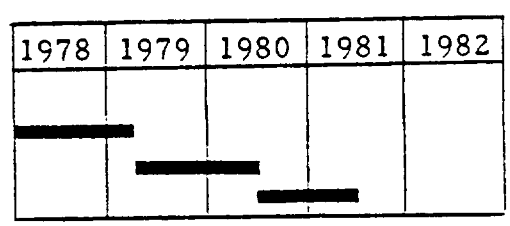
This effort includes review/development of recovery techniques most suited for NASA real-time space applications. This will be based on research into the structure and organization of software systems which provide reliable fault recovery through proper program organization methods. These developments should be coordinated with the activities proposed in other technology items, such as classification of error mechanisms (TI-17).

3.10.3 Suggested Approach

Structure analysis represents the first step of the proposed effort. This consists of identification of the types of faults likely to be encountered in real-time space systems and determination of the nature of the recovery process required to counteract these faults. Development of appropriate techniques and procedures for system recovery is the next step. This is followed by development of program organization methods which will maintain the status of the system so that the system can recover gracefully with minimum loss of data and control.

3.10.4 Schedule and Resources

- o Real-time program structure analysis.
- o Recovery techniques.
- o Program organization methods.



Estimated cost of development: \$200K - \$1000K.

3.10.5 Verification of Results

The development methods should be tested and evaluated by simulation under appropriate fault-inducing environments. This is extremely essential to prove the reliability of the techniques for deployment in real-world applications wherein the risk factors associated with improper recovery cannot be tolerated.

3.10.6 Related Studies

- Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-08 Interface criteria for NASA-distributed processor applications.
  - TI-09 Task control structures for distributed processor environments.
  - TI-12 Control structures for adaptive systems.
  - TI-15 Restructured image analysis software for parallel processing.
  - TI-17 Classification of error mechanisms.

3.11 Adaptive High-Speed Buffering Techniques for Dynamic Networks (TI-11)

3.11.1 Problem Definition

Missions involving image acquisition and other high-rate data-generating instruments, call for advanced capability in all areas of the data acquisition, handling, and processing system. The variable data rates of different experiments and the need for intelligent data acquisition dictates that advanced techniques, capable of adapting to changing data environments, be developed. High-speed buffering techniques in dynamic acquisition and switching networks are of prime importance, particularly in view of the high-speed, large-scale memory capability expected in the near future. Methods for optimal utilization of potential buffer resources will have tremendous impact on information gathering systems. This study represents a moderate to high cost developmental effort requiring 5-15 man years.

3.11.2 Technology Development Requirements

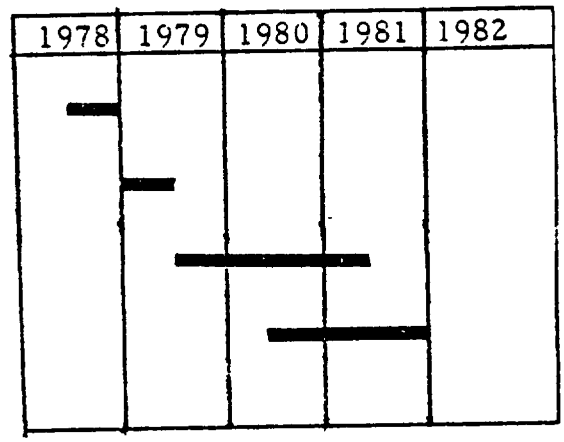
The technology development required by this item is the development of advanced high-speed buffering techniques capable of adapting to the dynamic characteristics of the data environment envisioned for future missions. These missions will involve high data rate experiments such as multispectral remote sensing of earth resources. This development is required to ensure that there will be an optimal utilization of the of the resources available both onboard and elsewhere in the total data system. This effort will contribute towards the overall objective of data system optimization by upgrading the state of the art in the key area of buffering techniques.

3.11.3 Suggested Approach

This developmental effort consists of several distinct steps as indicated under the schedule. The first step is a detailed assessment of the existing technology to identify deficiencies in the state of the art in meeting the future requirements. This requirements analysis must be supported by a parallel effort to identify the possible advancements expected in related hardware/software technology areas. With these efforts as the technology base, the study will proceed with the main task of developing new adaptive techniques. The possibilities of incorporating at least some basic level of intelligence at the front end will be explored. For example, when one is only interested in change detection, only data that are subject to change need be stored. The developed techniques will then be implemented, tested, and documented to ensure a sound return for NASA investment.

3.11.4 Schedule and Resources

- o Survey available techniques.
- o Identify advancements in related areas.
- o Develop new adaptive techniques.
- o Implement, test and document these techniques.



Estimated cost of development: $250K-750K.

3.11.5 Verification of Results

The developed techniques will be implemented and tested under properly simulated data environments including the expected range of dynamic data characteristics. This requires a thorough knowledge of the state of the art in data system simulation techniques in addition to those needed in the development phase of the effort. This verification of results, in terms of capability for meeting real-time requirements and other needs of the data system, will serve to enhance the value of the effort and help its acceptance by the user community.

3.11.6 Related Studies

o Software Technology Items.

- TI-19 Software prototyping methods.

3.12 Control Structures for Adaptive Systems (TI-12)

3.12.1 Problem Definition

Automated intelligence activities involve a variety of scene analysis, image processing and pattern recognition functions. These functions have to be called upon for processing, depending on the environmental conditions, in different sequences and configurations. This dependency requires the system to be adaptive to the needs of the changing environment. The problem is therefore to provide adaptive framework for the dynamic selection of the many software functions required in autonomous missions. The development of appropriate control structure concepts is expected to be of moderate cost requiring 8-12 man years.

3.12.2 Technology Development Requirements

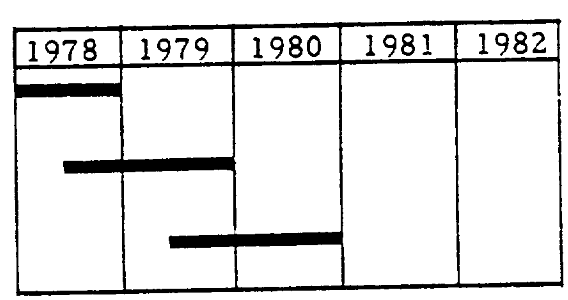
The technology advancement required is the development of control structure concepts for adaptive control of software procedures. The desired software system requires certain cognitive and decision making capabilities to adaptively deploy the right processing modules in an autonomously operating environment. The core development required encompasses conceptual, algorithmic and implementational aspects of the problem.

3.12.3 Suggested Approach

The initial step is to establish the requirements as stated above in more precise terms. A requirements analysis includes a review of the different processing functions involved in autonomous environments and their interrelationships. Once the specific processing functions and the possible configurations of their deployment are identified, it is then necessary to develop the concepts and techniques for adaptive control of these functions which can lead to optimal performance of the software system. This development of appropriate control structures will be followed up by experimental simulation (to check out the control structures under different test environments) and documentation to ensure its utility.

3.12.4 Schedule and Resources

- o Identify the interdependencies and relationships of software functions.
- o Develop appropriate control structures.
- o Test and document the developed control structures.



Estimated cost of development: $300K - $500K
Resource requirements include access to a computer system for test simulation of the developed control structures.

3.12.5 Verification of Results

A detailed simulation study plan must be formulated and implemented to ensure that the developed procedures meet their requirements. The simulation process should be carefully designed to provide an opportunity to test all feasible configuration options expected of the final software package.

3.12.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-14 Adaptive search and sort routines.
  - TI-15 Restructured image analysis software parallel processing.
  - TI-19 Software prototyping methods.

3.13 Impacts of Natural Communication Methods on NASA Payload Systems (TI-13)

3.13.1 Problem Definition

Multi-experiment missions call for a more optimal utilization of human resources. The control and monitoring of the different experiments should be spread, to the extent feasible, over all the different human faculties capable of interaction and communication. The computer system in control of these experiments must provide for man-machine communications and should make the best use of all known natural communication methods. The problem is the need to extend the scope of methods used onboard for man-machine interaction. This effort is expected to be one of relatively modest cost involving 3-6 man years.

3.13.2 Technology Development Requirements

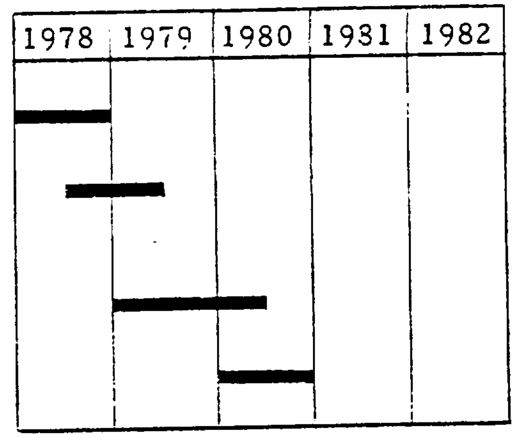
The goal of natural communication methods is best furthered by a study geared to assess their impact on the design of payload software systems. The assessment process must necessarily include a detailed analysis of typical implementations of such methods to develop a thorough understanding of existing potential. The study must critically evaluate the technology associated with these methods to clearly identify the feasible or potentially applicable methods which meet the NASA requirements for onboard implementation.

3.13.3 Suggested Approach

The starting point of this effort is a requirements analysis phase wherein the NASA requirements for natural communication methods in the upcoming missions are to be clearly identified and firmly established. Typical cases will be analyzed in depth through implementation/simulation to derive an understanding of their influence and potential impact on the supporting payload software system. This will lead not only to an evaluation of the feasibility of using these natural communication methods onboard, but also to an assessment of the necessary and desirable modifications to payload software system design. The results of such test case studies are then reviewed to derive an overall assessment of natural communication methods on space systems.

3.13.4 Schedule and Resources

- Identify NASA requirements for natural communication methods.
- Study the impact of these methods through analysis of typical implementations.
- Delineate necessary modifications to payload software system.
- Obtain an overall assessment of natural communication methods.



Estimated development costs: $150K - $250K.
Resources include the hardware for test implementation in a simulated onboard processing environment.

3.13.5 Verification of Results

The feasibility of employing the different natural communications methods will be verified by actual implementation in a simulated onboard processing environment. Such implementation will additionally lead to testing the supporting software system, thus enabling useful and reliable insight into all the different facets associated with natural communication techniques.

3.13.6 Related Studies

- Software Technology Items.
  - TI-08 Interface criteria for NASA-distributed processor applications.

3.14 Adaptive Search and Sort Routines (TI-14)

3.14.1 Problem Definition

Many of the pattern classification and image analysis techniques basic to an image processing system involve table look-up procedures operating on large arrays. These procedures necessarily call for search and sort operations in both the table formulation and look-up phases. The process is repeated innumerable times, especially in the look-up phase. This is because of the extensive volume of data handled by missions such as those involving earth resources survey and similar large imagery data acquisition experiments. It is therefore necessary to make such repeated table look-up operations as efficient as possible if images are to be processed onboard. Therefore, more efficient search/sort procedures are needed.

The problem facing the study is the need to overcome the possible inadequacies of presently known methodology in meeting the future onboard real-time processing system requirements. Further, it is desirable for these procedures to be adaptive to the array data characteristics to ensure optimum performance. It is expected that the required study is of relatively modest cost and will be approximately 3-6 man years of effort. This is because of the significant amount of related work reported in the area of search and sort techniques as applied to ground environments.

3.14.2 Technology Development Requirements

The technology development requirements can be precisely stated as the development of a set of search and sort routines and associated software control structures for adaptively selecting the appropriate routines for different data environments. This requires identification of the most appropriate methods for different array data characteristics. This will be done through review and modification of existing methods and/or additional developments followed by integration of these procedures within an adaptive structure.

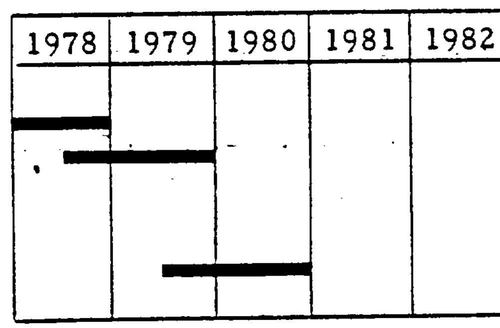
3.14.3 Suggested Approach

The first step of this development study is a detailed requirements analysis. This consists of a close scrutiny of the major pattern classification and image processing techniques to identify the specific needs for efficient search/sort procedures. These needs must be analyzed in the context of available information about the data rates, image sizes and such other related items pertaining to the data environment expected in future missions. This leads to a detailed definition of the requirements for search/sort routines under different identifiable classes of data environment characteristics as well as associated time constraints for possible real-time implementation.

The next step is to evaluate the available search/sort procedures in terms of their ability to meet these requirements and identify the potential candidates for adaptation. These candidate techniques are modified, improved and combined to enhance their potential. This effort may include the development of completely new methodology. However, such a need may not arise in view of the extensive work being performed on ground-based systems. The selected set of modified/developed techniques deemed optimal for the different data environments are then integrated into appropriate software control structures. These software control structures will be able to recognize the data environment characteristics and route the data flow through the appropriate software modules. The developed package will then be tested in simulated data environments to verify and validate the end product of the study.

3.14.4 Schedule and Resources

- Requirements analysis.
- Develop adaptive routines and integrate them through appropriate control structures.
- Implement, test and document the adaptive routines package.



Estimated development costs: \$150K - \$250K.
Resource requirements include access to a computer system capable of simulating the different data environments expected onboard.

3.14.5 Verification of Results

The developed set of procedures and their adaptive capabilities must be verified by simulation of the different data environments. This calls for appropriate simulation tools including a computer system of sufficient computational power. The verification process should be performed both at the level of individual routines to assess individual performances and at the integrated level to verify the adaptation capabilities of the total system.

3.14.6 Related Studies

- Software Technolcgy Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-12 Control structures for adaptive systems.
  - TI-15 Restructured image analysis software for parallel processing.
  - TI-16 Optimal large array partitioning procedures.
  - TI-19 Software prototyping methods.

3.15 Restructed Image Analysis Software for Parallel Processing (TI-15)

3.15.1 Problem Definition

Parallel processing is an attractive concept for the efficient realization of image processing and pattern recognition functions onboard NASA payloads in the coming decades. However, existing software techniques in the image analysis area are structured only for batch processing in conventional sequential processing environments on ground. Therefore, image analysis software must be appropriately tailored for implementation in parallel processing environments. The effort required for meeting the needs of this concept is expected to be moderately large in scope and to span 10-25 man years, depending on the number of software functions to be covered by the effort.

3.15.2 Technology Development Requirements

The enabling of onboard near-real-time image processing and pattern recognition activities requires full utilization of parallel processing technology to minimize the processing time required. This calls for operationally efficient software, structured to take advantage of the parallel processing environments feasible onboard in the next decade. Hence the major technology needed is the restructuring of image analysis software which is considered crucial to the enablement of image processing and pattern recognition activities onboard.

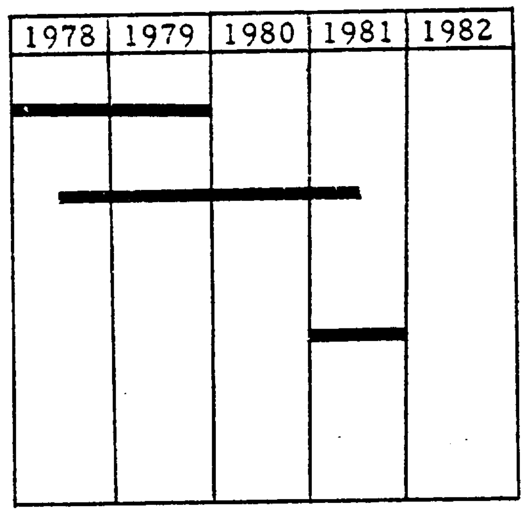
3.15.3 Suggested Approach

Parallelism can be visualized in terms of performing identical processing on different segments of an image or data array using parallel processing elements. It can also be visualized in the sense of carrying out independent computations on the same segment of data concurrently through different processing elements. It is therefore necessary to explore all such feasible parallelism in the major image processing functions of consequence to onboard implementation. For this purpose, it is also necessary to assess the state of the art in parallel processing techniques and supportive low cost hardware technology. This will help in determining the level or extent of parallelism advisable from a practical viewpoint.

Having identified the feasible and practical parallel processable functions and the key techniques, the software is accordingly restructured to achieve the maximum parallel processability. This is followed by identifying the appropriate computer architecture and control structure for implementing the restructured software system. In this context, the impact of developments in the area of optical and hybrid (optical and digital) processing should be assessed to ensure that the restructured system does not attempt to perform functions that are more efficient in optical processing techniques. This assessment must include the complexities of transforming images between digital to optical domains as that tends to offset the advantages of optical processing.

3.15.4 Schedule and Resources

- o Identify parallel processable functions.
- o Restructure/modify the software to achieve maximum parallel processability.
- o Identify the appropriate computer architecture for implementing the restructured software system.



Estimated development costs: $500K-$1000K

Resources include hardware for testing the restructured software in appropriate parallel processing environment.

3.15.5 Verification of Results

The results are to be verified through tests involving classes of multispectral data. Experiments should be performed with both the original and restructured software (the latter in parallel processing environment) to verify the correctness of the restructured software and to determine the computational gains achieved through parallel processing. This aids in verifying and validating the computer architecture and control structures to be used for purposes of onboard image processing.

3.15.6 Related Studies

- o Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.

- TI-05 Standardization criteria for NASA software.
- TI-08 Interfacing criteria for NASA distributed processor applications.
- TI-09 Task control structures for distributed processor environments.
- TI-12 control structures for adaptive systems.
- TI-20 Software prototyping methods.

3.16 Optimal Large Array Partitioning Procedures (TI-16)

3.16.1 Problem Definition

Manipulations of large two- and three-dimensional arrays which arise in the implementation of many data compression and image processing functions pose an operational problem that is far from trivial. Quite often the transposition of a matrix which is too large to fit in core, may require more time than performing basic transform or other operations on the rows and columns of this matrix. In addition, the complexity of a large number of input/output operations demands more efficiency in manipulating large arrays under the constraints of a real-time image analysis system.

This problem was identified in Technology Driver PS-15, Efficient Large Array Manipulation Procedures. A solution is expected to result from a study effort of relatively modest cost (3-6 man years).

3.16.2 Technology Development Requirements

The technology developments required to solve the problem defined above can be summarized as the development of optimal large array partioning procedures. The resulting procedures should be capable of adaptating themselves to the array data characteristics, i.e., the actual procedure used must be tailored to the array data characteristics expected in the corresponding environment. Currently known and reported methodologies for performing such matrix transpositions must be evaluated to pinpoint deficiencies in meeting the requirements of onboard real-time image processors. Partitioning of the arrays is a feasible approach in meeting the constraints expected. It is necessary to develop partitioning procedures which make the best use of the available computational resources and at the same time minimize time utilization. This technology development program shall study the impact of advancements in hardware technology such as electronic cyclic memories. This will ensure that the developed procedures are consonant with the state of the art in related fields, and that the resulting methodologies will meet the NASA requirements in the most efficient manner.

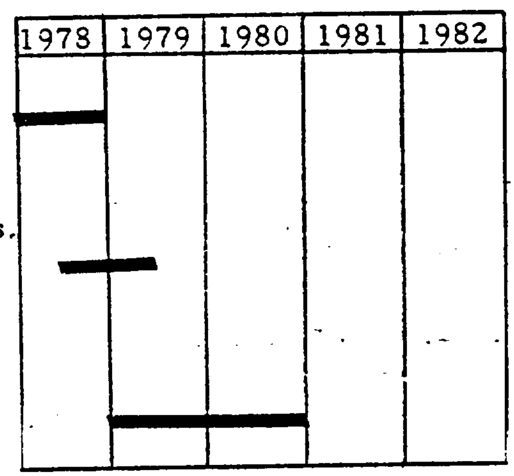
3.16.3 Suggested Approach

First, the study should consider the onboard processing system architecture (and constraints) expected by mid 1980's, and should restructure the optimal control problem accordingly. Second, alternatives to the dynamic programming approach should be explored.

Third, the feasibility of adapting the procedures to array data characteristics should be investigated. Fourth, the impact of technological advancements such as electronic cyclic memories on the manipulation procedures should be studied. The study should then be directed towards the development of suitable optimal partitioning procedures which can be applied to a variety of onboard processing environments. The deliverable end product items will be a set of strategies or optimal large array partitioning procedures which will lead to best utilization of the available onboard computational resources. In addition these procedures shall be adaptive to the array data characteristics.

3.16.4 Schedule and Resources

- o Study feasibility for adapting array manipulation procedures to array data characteristics.
- o Determine interrelationships of these procedures with data storage and processor architecture.
- o Develop, test and document these procedures.



Estimated development costs: $150K-250K. Resource requirements include access to a computer system capable of simulating different onboard processor environments.

3.16.5 Verification of Results

The methodology developed must be tested in different simulated onboard environments on a ground-based computer system to validate and verify optimality. The tests should be designed to cover a wide variety of array data characteristics to prove the adaptability of the resulting procedures to such characteristics. These tests will ensure the success of the task undertaken and can lead to advancement of the technology to the next level required for actual onboard implementation.

3.16.6 Related Studies

- Software Technology Items.
  - TI-05 Standardization criteria for NASA software.
  - TI-12 Control structure for adaptive system.
  - TI-14 Adaptive search and sort routines.

3.17 Classification of Error Mechanisms (TI-17)

3.17.1 Problem Definition

"Learning from History" is the most often suggested route for avoiding repetition of previously encountered errors. This is particularly valid in the case of software errors because of the lack of standardization in software design procedures and the consequent tendency to fall into the same errors repeatedly. The problem is therefore the need for development of avoidance techniques through a process of identification and classification of error mechanisms recorded in past software projects. NASA experience in space systems is particularly applicable. The effort required here is approximately 3-6 man years. The return, although perhaps difficult to measure, can be substantial in terms of cost avoidance.

3.17.2 Technology Development Requirements

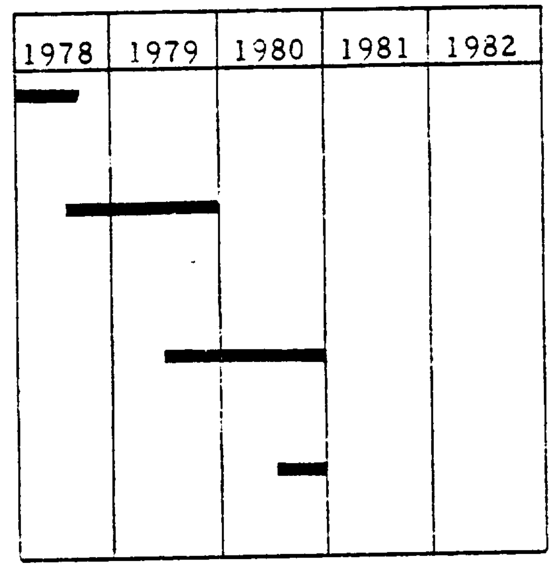
The technology developments required here are quite unlike others in that the end product is not so much a methodology but a directory categorizing the different types of software errors and the factors which lead to them. This categorization or classification of error mechanisms will be derived through analysis of prevalent errors encountered in prior space software development programs. Guidelines will thereby be developed to ensure that those pursuing future software development will be forewarned of potential faults.

3.17.3 Suggested Approach

The first step of the study is the compilation of errors documented under the different NASA software development projects of the past space programs. (This requires free access to such information by the organization conducting the study.) The compiled list should then be critically reviewed and grouped on the basis of common characteristics. An understanding of the causes of such errors will aid in developing classification of the error mechanisms. Guidelines will be developed to avoid these types of software errors in future design projects. This approach can be expected to enhance the search for fault-free software development in future NASA programs.

3.17.4 Schedule and Resources

- Compile error data of past NASA software program projects.
- Review and study the errors to extract their characteristics and model these errors into identifiable classes.
- Derive an overall comprehensive list of these error classes.
- Develop guidelines for avoiding such errors in future projects.



Estimated cost of development $150K - $250K.
Resources include unrestricted access to documented reports of experience of prior NASA software development programs.

3.17.5 Verification of Results

The concept of verification of results is not valid in the usual sense of the phrase as only future experience can prove the value of the guidelines developed from this process of classification of error mechanisms. However, a small scale software development program can be carried through in the light of these results and error avoidance costs estimated in a subjective manner. As in many research items, insight into the underlying causes of an effect can lead to unrelated advancements in technology.

3.17.6 Related Studies

This Technology Item relates to all software development technology areas.

3.18 Guidelines for Design Documentation Consistency (TI-18)

3.18.1 Problem Definition

An important aspect of software design process, which has received much attention, but few solutions, is the method of documentation of software design. No satisfactory method has been developed of conveying requirements into design representations which are understandable throughout the evolving responsibilities of a program life cycle. Attempts have been made to use data flow diagrams, functional representations, networks and other means. Though each may have specific advantages, none satisfies the need for unique clarity and understandability in a universal sense.

Inadequate details, lack of consistency and such other drawbacks in the documentation of the design can cause misinterpretation of the intent of the software designer by the programmer in the coding phase. This results in error-prone software with low reliability and consequent high costs in its maintenance. It is therefore clear that the lack of proper guidelines for design documentation represents a problem of sufficient importance to be addressed by NASA to ensure the developed software is space-worthy. The effort required to meet this problem effectively is expected to be one of relatively modest cost involving 2-5 man years.

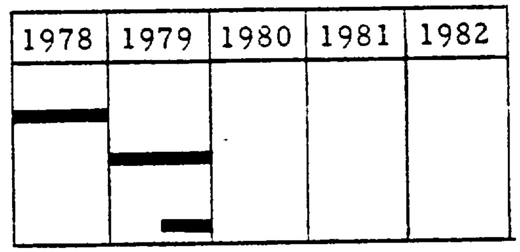
3.18.2 Technology Development Requirements

The developments expected of this study are the guidelines for consistency in design documentation which can facilitate clear understanding of the design coding requirements. This will enhance the scope for higher reliability in the generated code, thereby reducing maintenance. This is a desirable objective in space-related environments of concern to NASA. The guidelines should be broad enough in scope to cover the spectrum of NASA payload software activities and specific enough to be directly applied to the software development projects in the coming decade.

3.18.3 Suggested Approach

The effort should be closely coordinated with other software design-related Technology Items such as requirements decomposition and structuring guidelines, and AOL design guidelines/standards. The first step of this effort is to develop a clear understanding of the state-of-the-art design techniques (and hence the need to coordinate with efforts leading to improved design techniques). This will ensure that the design structure and all assoiated aspects are known and their implications are completely grasped prior to the development of documentation guidelines. This design structure analysis will be followed by formulation of the actual guidelines for achieving consistency in design documentation. The next step is the development of evaluation criteria to measure the effectiveness of these guidelines.

3.18.4 Schedule and Resources



- o Design structure analysis.
- o Development of documentation guidelines.
- o Evaluation criteria.

Estimated cost of development: \$100K - \$250K.

3.18.5 Verification of Results

The guidelines can be effectively verified as to their worth by application to real-world software design documentation efforts. The resulting documentation is assessed on the basis of the evaluation criteria developed as part of the study. This leads to a measure of effectiveness in achieving design documentation consistency.

3.18.6 Related Studies

- o Software Technology Items.
  - TI-01 Requirements decomposition and structuring guidelines.
  - TI-05 Standardization criteria for NASA software.
  - TI-06 AOL design guidelines/standards.

3.19 Software Prototyping Methods (TI-19)

3.19.1 Problem Definition

Software design errors and the associated costs are always a matter of grave concern to software development projects. Even more disconcerting is the discovery that a design will not meet requirements after implementation is complete. It is therefore highly desirable to develop methods for proving the software design before committing the design to code. This is the objective of software prototyping methods. Software prototyping will provide low-cost pretesting of the key elements of the design prior to implementation of the operational system. This technology when properly developed and used offers great benefits in reliability and cost avoidance. The development effort is expected to be one of relatively modest cost involving 2-5 man years.

3.19.2 Technology Development Requirements

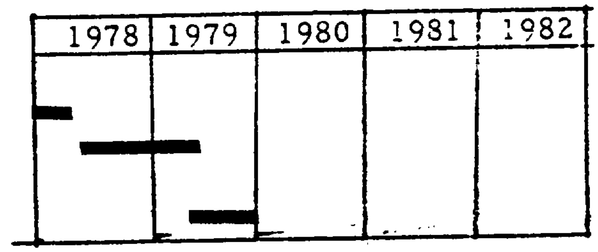
The technology to be developed under this study is the means of testing software design prior to implementation of the system. Such development will cut the high cost associated with design errors both in terms of their potential effects on reliability and in terms of the actual expenses involved in correcting these errors or making false starts. The methods to be used to achieve software prototyping capabilities represents the major requirement of this development study. Also, suitable evaluation criteria must be developed to assess the effectiveness of these methods.

3.19.3 Suggested Approach

The effort must be mainly directed towards the development of methods suitable for prototyping real-time space systems. It is necessary to assess the technology currently available in the area of ground-based prototyping techniques prior to exploring the scope for prototyping onboard real-time systems. The current technology analysis will then provide pointers for extrapolation of these techniques into the domain of space. This extension may not be straightforward in that the environmental constraints associated with space systems are likely to be of an entirely different category. The final stage of the effort is the development of appropriate evaluation criteria for assessing the developed techniques.

3.19.4 Schedule and Resources

- o Technology analysis.
- o Low-cost prototyping methods.
- o Evaluation criteria.



Estimated cost of development: $100K-250K.

3.19.5 Verification of Results

The end product of this effort and its effectiveness can be judged by application to specific cases of software design and development programs. The evaluation criteria developed as part of this technology development study will be used to assess the gains and verify the efficiency of the tools resulting from the study. Comparative analysis with non-prototype development will be performed where systems with similar requirements can be identified.

3.19.6 Related Studies

This Technology Item is applicable to all software development technology.

3.20 Software Technology Monitoring (TI-20)

3.20.1 Problem Definition

NASA should capitalize on the continuous evolution and the occasional breakthroughs occurring in the areas of hardware/software technology. A process of early awareness of the developments is needed followed by a study of the impacts of such developments on NASA programs. This requires that there be a constant monitoring of the developments in the field and for this purpose it is necessary that NASA establish a suitable formal mechanism.

As an Item, software technology monitoring seems obvious and deceptively simple. NASA personnel are continually monitoring technology and providing much of the advancement. However, this monitoring basically applies to a specific area of interest of the person or group doing the monitoring. The key to this task is to have a central source monitoring all software/hardware technology and correlating advancements to ongoing or proposed space activities wherever they may be applicable. This is to be an on-going low level effort of relatively low cost, requiring 1 man year per year.

3.20.2 Technology Development Requirements

In this case, the requirement is not truly one of technology development but rather one of technology assessment. However, the requirements can be viewed as developing and instituting a formal software technology monitoring mechanism. Such a mechanism would be responsible for performing the monitoring activity and correlating developments with the NASA requirements. It would thereby contribute towards identification of additions (and/or deletions) to the NASA Software Technology Item roster.

3.20.3 Suggested Approach

The formal mechanism set up for this purpose should have access to information about activities from different sources within NASA and outside of it also, to the extent feasible. This is in addition to the open literature publications which are easily accessible. The formal mechanism should utilize the modern techniques of technology forecasting and assessment, such as advanced DELPHI techniques, not only for monitoring but also to identify the expected course of developments. This activity being an on-going effort, the projections can be verified whenever promising technology is identified.

3.20.4 Schedule and Resources

	1978	1979	1980	1981	1982	- -	1986
o Monitor software technology advances.	— —	— —	— —	— —	— —	—	— —

Estimated development costs: $50K/year.

3.20.5 Verification of Results

This Technology Item is significantly different from others; therefore the concept of verification results does not apply in its usual sense. However, the projections on future developments made earlier can be verified by the monitoring process, thereby refining the assessment methodology.

3.20.6 Related Studies

Not applicable (except in the sense that the Technology Item is related to the total area of software technology).

3.21 Spacelab Software Technology Experiment (TI-21)

3.21.1 Problem Definition

We do not yet know how software can best support man in space, nor what capabilities exist to relieve the burden of a flight crew, to make space flight more productive, more reliable and less expensive. Questions such as, what are the most effective display techniques? how can graphics be used? how to detect targets-of-opportunity? how to improve data quality or select data in real time?, have not been pursued in a real environment under mission constraints.

Spacelab Mission I presents an opportunity to actually advance software technology through the testing of techniques, concepts and applications in a real environment. This is an opportunity that may not present itself again. Resources will be available on the CDMS subsystem computer to allow experimentation with passive software packages in a non-interference mode. An exercise of this type will provide invaluable insight into how software can best interface with a flight crew to maximize mission performance.

3.21.2 Technology Development Requirements

Provide the passive software packages necessary to seek technology enablement advancements through the use of available resources in a non-interference mode in Spacelab Mission I and subsequent Spacelab missions.

3.21.3 Suggested Approach

The initial task of this study is to conduct a feasibility exercise supporting the concept and identification of high-potential software packages for flight test. This software must be passive and must not interfere in any way with mission flight software. Typical candidate software experiments might be:

- Data Compression/Decompression Effects on Image Reconstruction.
- Evaluation and Selection of Data Display/Interface Techniques in a Real, Onboard Environment.
- Demonstrate Use of AOL for Real-Time Software Function Generation/Modification Onboard.

o Use of Onboard Computers/Software as an Engineering/ Scientific Test Tool.

- Slide-Rule Calculator.
- APL.
- Graphic Generation.
- Scratch Pad (Text-Drawings, Activity Log, etc.).

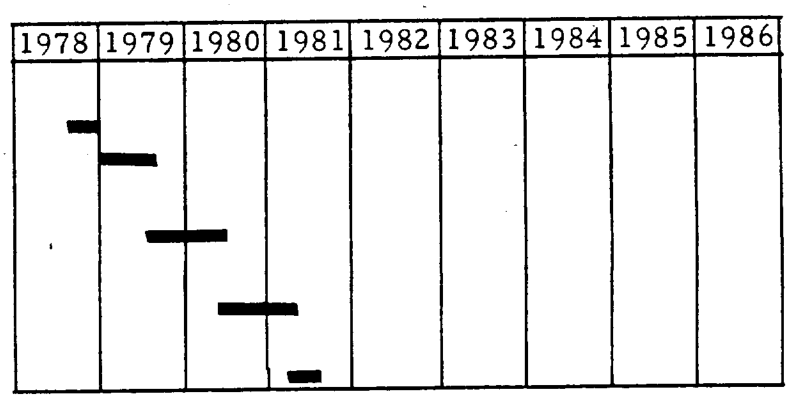
o Evaluation of Natural Communication Methods.

Completion of the feasibility test will provide data necessary to proceed with the software package design and prototype development and test. Successful testing of the prototype should be followed by appropriate Mission I crew training and the actual flight test of the software experiment.

Post flight evaluation will be a key milestone in determining the next steps in technology advancement and will also serve as a test to determine if follow-on experimentation should be initiated for subsequent flights.

3.21.4 Schedule and Resources

o Feasibility Study.
o Software Package Design.
o Prototype Development and Test.
o Crew Training & Flight Test.
o Post Flight Evaluation.



Estimated Cost of Development: \$100K - \$300K

Resources must include access to the Spacelab CDMS subsystem computer and display system.

3.21.5 Verification of Results

Four key milestones are available to test intermediate and final results:

o Feasibility study results.

o Prototype test results.

- Crew-training response.
- Post-flight evaluation.

3.21.6 Related Studies

TI-08 Interface Criteria for Distributed Processor Environments.

TI-13 Impacts of Natural Communication Methods on NASA Payloads.

TI-19 Software Prototyping Methods.

3.22 On-Board Radiometric Correction (TI-22)

3.22.1 Problem Definition

Ground systems are large and expensive. A key to continued growth of remotely sensed data capability is continued reduction in the cost of the data collected; therefore, it is critical that the cost of ground processing be reduced. New hardware technology has made this possible by opening the door to significant onboard processing. A re-evaluation of processing distribution is now necessary to take full advantage of the potential cost-savings offered. Costs to the user of remotely sensed data must be reduced in order to generate broad and general support of space programs. Redistribution of sensor dependent functions to the sensor system can reduce ground processing requirements.

The major problem has been, and probably will continue to be, a general reluctance to add additional hardware or processes in serial with the data stream, in an onboard environment, for fear of data loss or degradation. Although an understandable, and probably valid, consideration in the past, this concern should be re-evaluated in light of the new technology available now to build in reliability and redundancy at very little impact to payload cost, weight or volume.

3.22.2 Technology Development Requirements

A general reassessment of software functional distribution is required. Specifically, those functions that are sensor dependent should be studied for potential application to a "smart" sensor. Possible functions are:

- o Radiometric correction.
- o Quality assessment.
- o Cloud cover detection.
- o Change detection.

As a first step, it is recommended that onboard radiometric correction be considered, and a prototype system developed.

3.22.3 Suggested Approach

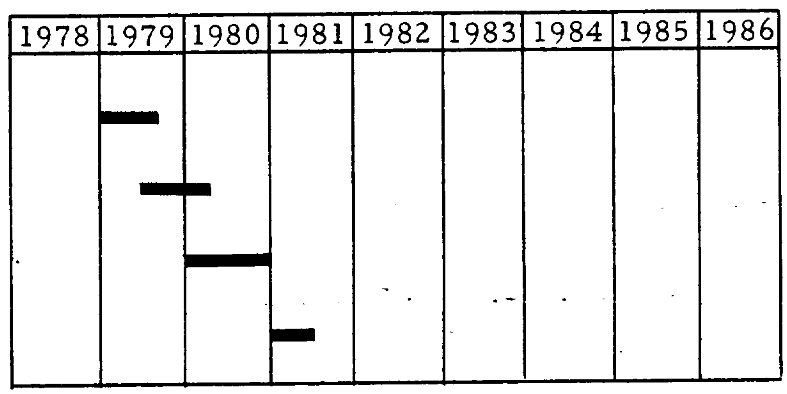
Development of a prototype onboard processing system to perform radiometric correction of remotely sensed data is feasible. The first step should be a complete reassessment of the functional distribution of processing requirements to ensure that radiometric correction by itself is the prime candidate for redistribution, or if it should be accompanied by various other functions.

The second step should be the definition and design of a prototype software system. The third step encompasses development and test of the prototype system after completion of a thorough design review.

The final step, prior to development of an operational system, must be an exhaustive test of system reliability and recovery features. This step is critical because the results can hold significant implications for the future of onboard processing.

3.22.4 Schedule and Resources

- o Reassess Functional System Distribution.
- o Design Prototype System.
- o Develop and Test Prototype.
- o Perform Reliability/ Recovery Assessment.



Estimated Cost of Development: $150K - $250K

Resources must include access to a non-flight qualified payload processor and imaging sensor simulator.

3.22.5 Verification of Results

Verification of results would take two forms:

(1) Comparison of image results with equivalent ground based radiometric correction.

(2) Demonstration of the capability to recover from induced faults with minimum loss or degradation of data.

3.22.6 Related Studies

TI-01 Decomposition and Structuring Guidelines.

TI-08 Interface Criteria for NASA-distributed Processor Applications.

TI-09 Task Control Structures for Distributed Processing Systems.

TI-11 Adaptive High-speed Buffering Techniques for Dynamic Networks.

3.23 Onboard Classification and Distribution (TI-23)

3.23.1 Problem Definition

Data distribution poses one of the greatest impediments to full realization of the potential benefits of space-borne remote sensing technology. The cost of ground processing and data communication must be reduced. This can be accomplished perhaps through direct transmissions to the user, perhaps through ubiquitous, low cost processing systems, or perhaps through communications breakthroughs, but in any event, must result in maximum benefit to the user at minimum cost.

3.23.2 Technology Development Requirements

Development of an onboard, image classification system to generate and distribute moderate resolution classification maps for wide area, real-time reception by end-users. This system would operate in parallel with the prime data stream on a non-interference basis and would be used only to augment or supplement the established data collection/processing/ distribution system.

3.23.3 Suggested Approach

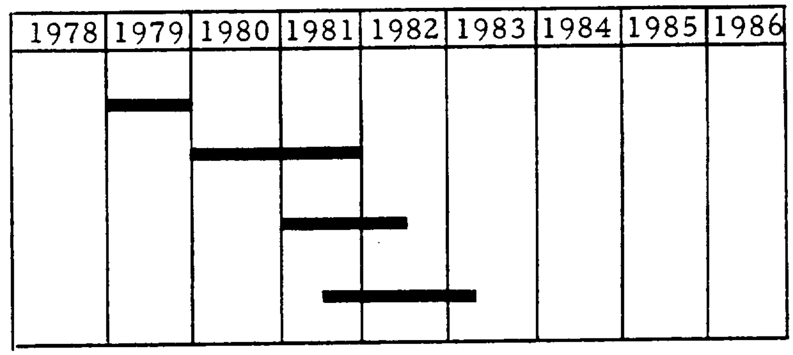
Prototype image classification systems for ground processing have been developed, e.g., the Multivariate Interactive Digital Analysis System (MIDAS) developed for NASA by the Environmental Research Institute of Michigan. These systems should be thoroughly analyzed for applicability to onboard implementation. This analysis should be accompanied by a complete onboard system definition.

Next, a prototype system should be developed and tested. This must be accompanied by the concurrent development and test of associated user terminals and antenna systems. The over-riding concern of user terminal design must be low cost.

On satisfactory completion of prototype system test, an operational prototype (flight qualified) should be deployed for on-orbit testing.

3.23.4 Schedule and Resources

- o System definition.
- o Prototype development and test.
- o User terminal/antenna system development.
- o Deployment of an operational prototype.



Estimated Cost of Development: $250K - $500K, exclusive of hardware.

3.23.5 Verification of Results

Verification can be readily accomplished by comparison of results achieved on a large-scale ground image classification system, using duplicate image data.

3.23.6 Related Studies

TI-08 Interface Criteria for NASA-distributed Processor Applications.

TI-11 Adaptive High-speed Buffering Techniques for Dynamic Networks.

TI-15 Restructured Image Analysis Software for Parallel Processing.

3.24 Security and Control of High Resolution Data (TI-24)

3.24.1 Problem Definition

High resolution sensors have created two immediately recognizable problems: (1) protection of military significant data and (2) protection of rights to privacy. Data collected from these sensors is vulnerable during transmission from point-to-point and is also vulnerable when it resides in either temporary or permanent storage devices. This item provides for the development of software approaches to the identification and safeguarding of this data.

3.24.2 Technology Development Requirements

There are two major facets of this problem. The first is the development of techniques to recognize sensitive data. The second is the development of tools to protect the sensitive data after it is recognized.

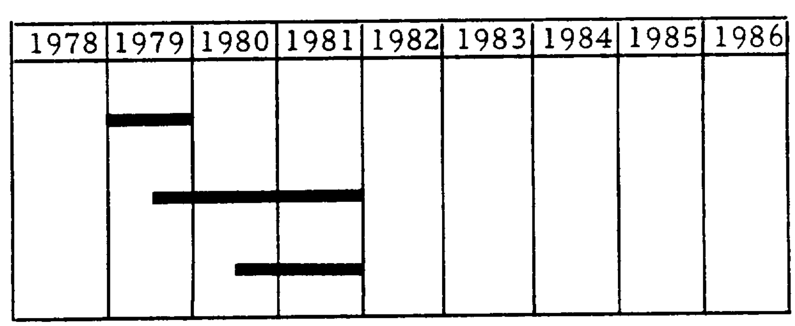
3.24.3 Suggested Approach

It is recommended that a study be undertaken to determine methods of identifying sensitive data, and once this has been completed, build and test feasibility models of software systems to monitor and control this data. Alternative to software systems exist such as a priori selection of data to be collected or various hardware schemes; however, in either case, software will play a part.

The final step should be development of methods to monitor and report access to all data.

3.24.4 Schedule and Resources

- o Identification of sensitive data.
- o Development and test of feasibility models.
- o Development of access monitoring techniques.



Estimated Cost of Development: $250K - $500K

Access to a large scale computer will be required to exercise simulations.

3.24.5 Verification of Results

Verification of this task will be rather difficult. It will depend on the availability of qualified experts in the data analysis/data security field to make objective judgments as to the validity of the identification/protection mechanisms.

3.24.6 Related Studies

TI-05 Standardization Criteria for NASA Software.

TI-12 Control Structures for Adaptive Systems.

TI-14 Adaptive Search and Sort Procedures.

3.25 Low-Cost User Systems (TI-25)

3.25.1 Problem Definition

Current technology centers around space-to-earth transmission to central facilities such as TDRSS/White Sands. Earth-to-earth technology is primarily relegated to commercial satellite communications advancement. However, the link from source to end-user is still complex. Direct links have not been pursued to a necessary depth. Low cost distribution of data to end-users of remotely sensed data is critcal to the future of NASA space programs. Ground distribution networks are expensive and, in many cases, slow. New methods of distribution must be developed.

3.25.2 Technology Development Requirements

Low-cost user systems for final processing of data and enablement of direct transmission of onboard preprocessed data is key to the development of a broad-based user community and reduced distribution costs. Millions of dollars in redundant RID could be saved if standard systems (including software) were developed by a central agency such as NASA.

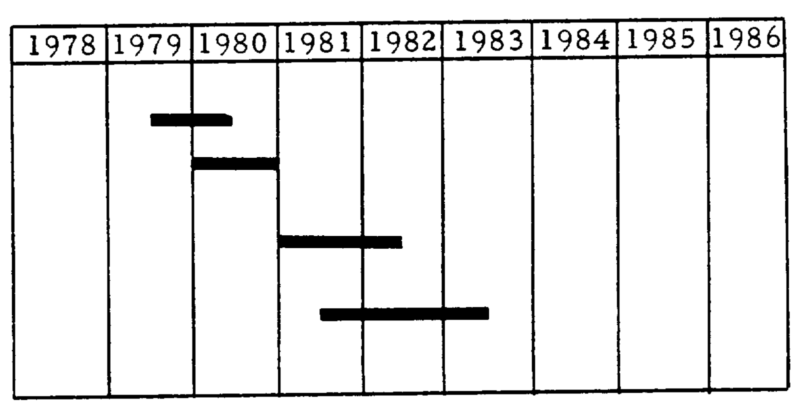
3.25.3 Suggested Approach

The initial step in this task must be a complete definition of user needs and user processing requirements. The next step requires characterization of hardware components to satisfy these requirements.

After the initial steps are thoroughly complete, hardware systems must be procured and integrated. Subsequently, software systems can be implemented and a library of tested software applications stored and maintained in an accessible software library for distribution to end-users.

3.25.4 Schedule and Resources

- o Identify user needs.
- o Establish system requirements.
- o Develop hardware system.
- o Define build, and test software system/ applications.



Estimated Cost of Development: $250K - $500K, exclusive of hardware procurement.

3.25.5 Verification of Results

Complete support and participation of users will have to be employed in the development of requirements. Results will be verified by the proven development of a low-cost system that performs to the level specified by initial requirements.

3.25.6 Related Studies

TI-05 Standardization Criteria for NASA Software.

TI-06 AOL Design Guidelines/Standards.

TI-07 AOL Compiler Generator Cost Factors.

3.26 Earth Model Data Bank (TI-26)

3.26.1 Problem Definition

Data collection from space is relatively expensive; time-on-orbit is precious. The problem is how to preserve the data collected until all beneficial information has been retrieved from it and, at the same time, support a retrieval system that is low-cost. The key is learning how to apply the new mass storage device technology in the most cost-effective manner to preserve, protect and provide simple access to the tremendous data volume generated by future space systems.

Many methods of storing and retrieving data are now available. These have been used for individual pieces of data (random image frames, specific mission return, etc.), but there has not been an attempt to build a comprehensive representation of the earth in a data bank that maintains the current state of the earth's surface in a universally accessible form. Acquisition of earth and ocean data has been solved. Access to this data has not and needs to be.

3.26.2 Technology Development Requirements

Development of a mass data bank which contains the current state of the earth's surface at various levels of resolution, in a form that is universally accessible, and within a structure that provides rapid retrieval of specific data within defined geographical areas at low cost and to the resolution required. The hardware technology exists; data base structure methodology is mature. Only the effort to get it-all-together remains. This effort would be of inestimable benefit to NASA in the pursuit of operational earth and ocean resource programs.

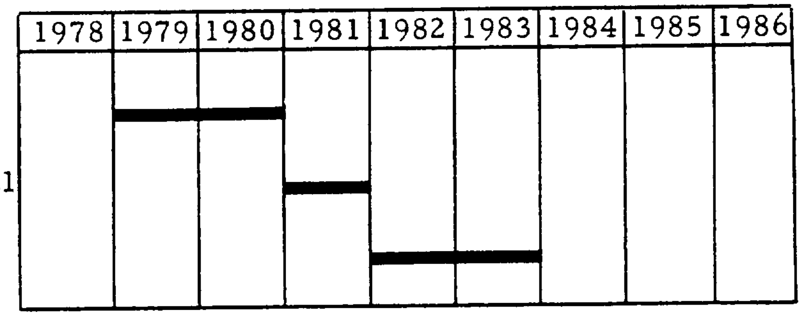
3.26.3 Suggested Approach

There are three steps required to achieve the goals of this item. These are:

- Identify data base form, content, and structure.
- Define the storage/retrieval protocol.
- Establish and maintain a current earth model.

3.26.4 Schedule and Resources

- o Identify data base structure.
- o Define storage/retrieval protocol.
- o Establish and maintain earth model.



Estimated Cost of Development: $1.5M - $3M.

3.26.5 Verification of Results

Verification basically depends on maintaining assurance of feasibility throughout the development process.

3.26.6 Related Studies

TI-14 Adaptive Search and Sort Procedures.

TI-16 Optimal Large Array Partitioning Procedures.

3.27 Scene Content Extraction (TI-27)

3.27.1 Problem Definition

There is a need for surface rover vehicles to be able to maneuver without direct manned supervision. In order to do this, the surface rover requires a semblance of vision in order to select a destination and plan an optimum course to the destination which will avoid obstacles and pitfalls. This requires the ability to extract information from a scene detected by a sensor (vidicon, CCD, etc.) in order to identify and categorize objects within that scene. Although the capability exists, in software, to do this with large ground based systems, technology advancements must be made in order to perform this task within the limits of the resources available to a planetary spacecraft and do so rapidly enough to make the rover concept feasible. Solution to the problem posed by this driver is mandatory to enable the surface rover concept.

3.27.2 Technology Development Requirements

The technology development required for this item, is the creation of scene analysis techniques which will allow for rapid, accurate extraction and identification of significant objects within a scene. These techniques must fit within the resource capacity allocated to planetary spacecraft and meet the fast run-time requirements ($\cong$ <3 minutes/scene) essential to efficient surface transit.

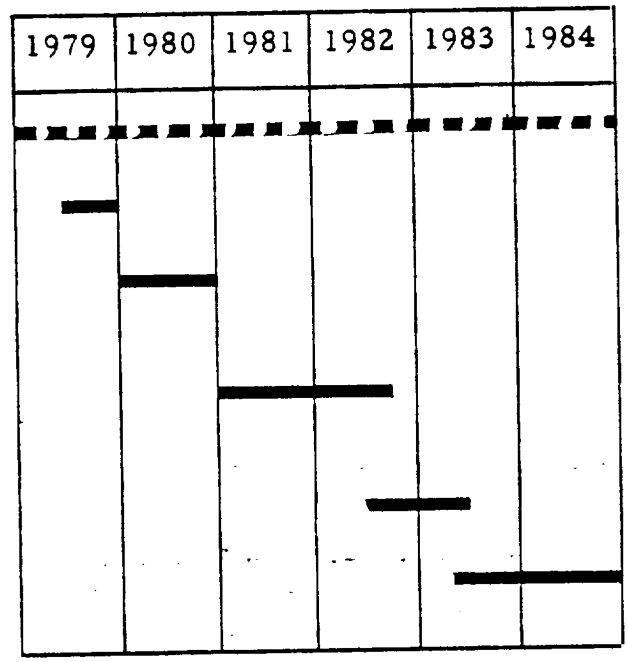
3.27.3 Suggested Approach

The R&D effort proposed for this topic consists of six steps as follows:

First, survey and identify major alternative approaches presently being explored in the area of scene analysis. Next, identify feasible candidates for further indepth evaluation in the context of applications, such as robotics, oil slick detection, snow cover delineation, etc., and related computational resource constraints. In the third step, evaluate segmentation of natural scenes to identify objects and derive information as to their location and orientation. (Possibilities of extending these to include man-made objects could also be looked into if needed.) Fourth, investigate the scope of integrating these candidate techniques to obtain a practical scene analysis technology compatible with available computational resources - for example, cluster analysis approach of grouping elements (sub-images) of the scene can be possibly integrated with edge detection techniques in developing appropriate merge/linkage strategies to derive a viable number of image segments. Fifth, extend these techniques to analysis of three dimensional (real outdoor) scenes by using images obtained from different perspectives. And last, derive the hierarchical tree of the efforts required to develop a practical scene analysis system to suit NASA program needs of the next two decades - for example, development of a simple first level partitioning or image segmentation strategy.

3.27.4 Schedule and Resources

- o Survey Major Approaches.
- o Identify Feasible Candidates.
- o Evaluate Segmentation Techniques.
- o Integrate Scene Analysis Methodology.
- o Extend Method to 3-dimensions.
- o Derive Hierarchy of Practical Scene Analysis Systems.



Estimated Cost of Development: $400K - $600K

3.27.5 Verification of Results

Verification of results will require: (1) demonstration of extraction and identification accuracy of the scene analysis system(s) using typical scenes; (2) verification of the run-time requirements of the system(s) against diverse scene characteristics (i.e., homogenous, geometric, high contrast, low contrast, noise); and (3) full up tests of the selected system(s) on a prototype rover vehicle in a real-life environment.

3.27.6 Related Studies

- o Software Technology Items
  - TI-12 Control Structures for Adaptive Systems.

- TI-14 Adaptive Search and Sort Procedures.

- TI-15 Restructured Image Analysis Software for Parallel Processors.

- TI-28 Path Planning Techniques.

3.28 Surface Navigation and Mapping (TI-28)

3.28.1 Problem Definition

The second major requirement of software for surface roving vehicles (after scene analysis) requires selection of a destination, planning an optimum course to that destination and being able to recognize the destination when it is reached. These are not trivial problems. Current solutions require massive secondary storage for the manipulation of large data arrays and powerful processors to do the mapping function. New algorithms must also be developed to predict the changes in scene content due to a change in perspective as the vehicle moves toward its destination. To perform these functions with the limited resources available to an autonomous spacecraft, will require extensive breakthroughs in software technology if the surface rover concept is to become feasible.

3.28.2 Technology Development Requirements

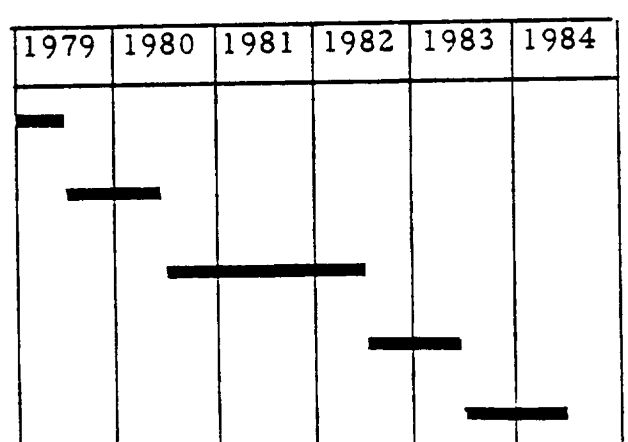
The requirements include development of algorithms to partition and manipulate large data sets without ready access to massive, rapid-access secondary storage and development of the techniques needed to predict changes in a scene that result from a change in perspective. At the same time, program run-times must be kept within limits which permit feasible operation of autonomous vehicles.

3.28.3 Suggested Approach

The first step in this development task must necessarily be a review of the state-of-art of robotics and roving vehicles. This must be followed by a thorough mathematical analysis of the problems posed and development of algorithms to solve these problems. The algorithms will then be tailored to flight-type processing systems. The final steps will involve extensive testing of the algorithms in simulated and real environments.

3.28.4 Schedule and Resources

- o State-of-Art Survey
- o Mathematical Analysis
- o Algorithm Development
- o Simulated Environmental Tests
- o Real Environmental Tests



Estimated Development Cost: $300K - $600K

3.28.5 Verification of Results

The developed techniques can initially be tested using simulated environments. The initial tests must verify that path selections are near optimum and that hazardous terrain can successfully be avoided. The next test must demonstrate that program run-time is within prescribed limits and that baseline spacecraft processing resources are not exceeded.

The final (or validation) tests will prove the techniques under rigorous exercises in real terrain using flight-like systems.

3.28.6 Related Studies

- Software Technology Items
  - TI08 Interface Criteria for NASA Distributed Processor Applications.
  - TI09 Task Control Structures for Distributed Processor Environments.
  - TI12 Control Structures for Adaptive Systems.
  - TI14 Adaptive Search and Sort Procedures.
  - TI16 Optimal Large Array Partitioning Procedures.
  - TI27 Scene Content Extraction.

(BLANK)